UCI-AM-16-009

ACQUISITION RESEARCH PROGRAM SPONSORED REPORT SERIES

Achieving Better Buying Power through Acquisition of Open Architecture Software Systems Volume I

6 January 2016

Dr. Walt Scacchi

Dr. Thomas A. Alspaugh

Institute for Software Research University

University of California, Irvine

Approved for public release; distribution is unlimited.

Prepared for the Naval Postgraduate School, Monterey, CA 93943.

The research presented in this report was supported by the Acquisition Research Program of the Graduate School of Business & Public Policy at the Naval Postgraduate School.

To request defense acquisition research, to become a research sponsor, or to print additional copies of reports, please contact any of the staff listed on the Acquisition Research Program website (www.acquisitionresearch.net).

Abstract

This research focuses on continuing investigation and refinement of techniques for identifying and reducing the costs, streamlining the process, and improving the readiness of future workforce for the acquisition of complex software systems. Emphasis was directed at identifying, tracking, and analyzing software component costs and cost reduction opportunities within the acquisition life cycle of open architecture (OA) systems for Web-based and mobile devices, where such systems combine best-of-breed software components and software products lines (SPLs) that are subject to different IP license and cybersecurity requirements. The investigation focuses on four project work activities:

- Investigating the interactions between software system acquisition guidelines and processes, and the cost consequences of alternative software system architectures incorporating different mixes of OSS and CSS components subject to different licenses within secure OA SPLs [ScA08, ScA12b, ScA13a, ScA13b, ScA13c]. This entails exploring the balance between development, verification, and validation of software licenses and security rights, as well as the software component/license costs while managing the development and evolution of OA systems at design-time, build-time, and release and run-time.

- Developing formal foundations for establishing acquisition guidelines program managers can use in reduced cost acquisition of software-intensive systems that rely on development and deployment of secure OA systems using OSS and SPL technology and processes [AIS10, AIS13, ScA11, ScA12a, ScA12b, ScA13a, ScA13b, ScA13c].

- Continuing to develop concepts contributing to the emerging design of an automated approach supporting acquisition of secure OA systems by (a) determining their conformance to acquisition guidelines/policies, contracts, and related license management issues, and (b) giving future acquisition workforce support and insights to properly review, approve, and manage the acquisition of complex systems that incorporate cost-sensitive acquisition of OA systems and software components [AIS10, ScA11, ScA12a, ScA12b, ScA13a, ScA13b, ScA13c].

- Documenting the investigation, foundations, and results of the research in: (a) a technical Final Report delivered to the Technical Point of Contact at NPS; (b) a research presentation at the 11th Annual Acquisition Research Conference, in Monterey, CA, May 2014; (c) a progress report with the OSD sponsor and others of interest within the OUSD (AT&L) offices; and (d) related research venues and publications, including periodic research progress reports.

THIS PAGE LEFT INTENTIONALLY BLANK

About the Authors

Dr. Walt Scacchi is a senior research scientist and research faculty member at the Institute for Software Research, University of California, Irvine. He received a PhD in information and computer science from U.C. Irvine in 1981. From 1981 to 1998 he was on the faculty at the University of Southern California. In 1999 he joined the Institute for Software Research at U.C. Irvine. He has published more than 150 research papers and has directed 60 externally funded research projects. In 2011 he served as co-chair for the 33rd International Conference on Software Engineering - Practice Track, and in 2012 he served as general co-chair of the 8th IFIP International Conference on Open Source Systems.

Dr. Walt Scacchi, Senior Research Scientist
Institute for Software Research
University of California, Irvine
Irvine, CA 92697-3455 USA
E-mail: wscacchi@ics.uci.edu

Dr. Thomas Alspaugh is a project scientist at the Institute for Software Research, University of California, Irvine. His research interests are in software engineering, requirements, and licensing. Before completing his PhD, he worked as a software developer, team lead, and manager in industry, and as a computer scientist at the Naval Research Laboratory on the Software Cost Reduction or A-7 project.

Dr. Thomas Alspaugh, Project Scientist
Institute for Software Research
University of California, Irvine
Irvine, CA 92697-3455 USA
E-mail: thomas.alspaugh@acm.org

THIS PAGE LEFT INTENTIONALLY BLANK

UCI-AM-16-009

ACQUISITION RESEARCH PROGRAM SPONSORED REPORT SERIES

Achieving Better Buying Power through Acquisition of Open Architecture Software Systems Volume I

6 January 2016

Dr. Walt Scacchi

Dr. Thomas A. Alspaugh

Institute for Software Research University

University of California, Irvine

Disclaimer: The views represented in this report are those of the author and do not reflect the official policy position of the Navy, the Department of Defense, or the federal government.

THIS PAGE LEFT INTENTIONALLY BLANK

Executive Summary

The goal of this research was to create a new approach to address Better Buying Power challenges in the acquisition of software systems for the Department of Defense. Program managers, acquisition officers, and contract managers will increasingly be called on to review and approve choices between functionally similar low or no cost open source software components, and commercially priced closed source software components, to be used in the design, implementation, deployment, and evolution of open architecture (OA) systems. We seek to make this a simpler, more transparent, and more tractable process. Such a process must identify, track, and analyze software component costs throughout the system life cycle, and be easy to reuse for different system application domains, in order to realize cost reductions and improve acquisition workforce capabilities. Our recent research demonstrates how complex OA systems can be designed, built, and deployed with alternative components and connectors resulting in functionally similar system versions, to satisfy overall system security requirements and individual system component intellectual property (IP) requirements [DODOSA13, SEI13]. Our next step, described in this two volume Final Report, is to identify, track, and analyze software component costs associated with different types of component IP licenses when acquiring OA systems, and to do so in ways that highlight opportunities for cost reduction. We believe our results will be applicable to enterprise software systems in other government agencies and industrial firms, as well as to enterprise and mission-critical systems for the DoD community.

This research focuses on continuing investigation and refinement of techniques for identifying and reducing the costs, streamlining the process, and improving the readiness of future workforce for the acquisition of complex software systems. Emphasis was directed at identifying, tracking, and analyzing software component costs and cost reduction opportunities within the acquisition life cycle of open architecture (OA) systems for Web-based and mobile devices, where such systems combine best-of-breed software components and software products lines (SPLs) that are subject to different IP license and cybersecurity requirements.

The Department of Defense, other government agencies, and most large-scale business enterprises continually seek new ways to improve the functional capabilities of their software-intensive systems. The acquisition of OA systems that can adapt and evolve through replacement of functionally similar software components is an innovation that can lead to lower cost systems with more powerful functional capabilities. This research seeks to identify and analyze how software component costs for Web-based and mobile devices, component IP license and cybersecurity requirements interact to drive down (or drive up) total system costs across the system acquisition life cycle. The availability of such new scientific knowledge and technological practices can give rise to more effective expenditures of public funds and improve the effectiveness of future software-intensive systems used in government and industry. Thus, the principal purpose of this research supports and advances a public purpose.

Finally, our principal research results are documented in two volumes.

Volume I includes four contributions. In Chapter 1 we summarize details of our research efforts in the past 12 months. These efforts have been well received in presentations to different audiences, including within the larger Defense community, and the Federal Government more broadly. In particular, our research results have been picked up for use within the *Assembled Capabilities Working Group* (ACWG, previously identified as the DoD Widget Working Group, through early 2014), under the guidance of the C3CB (Command, Control, Communications, and Business Systems) office within the OUSD (AT&L). This effort was facilitated through collaboration with many people from The MITRE Corporation, who along with the C3CB office are working in support of the Defense Intelligence Information Enterprise (DI2E) and related mission partners. Summary presentations that have been publicly shared resulting from our research appear in Chapter 2, 3, and 4. Chapter 2 includes the abstract and slide deck that were presented at the 2014 Acquisition Research Symposium

(May 2014). Chapter 3 is the slide deck from MITRE-ATARC Workshop in Washington, DC (August 2014) addressing Cost-Sensitive Acquisition of Open Architecture Software Systems for Mobile Devices. Chapter 4 is the slide deck from the Federal Mobile Computing Summit also held in Washington, DC (August 2014). Further, in response to many requests for additional information on our research approach, methods, and results, we have compiled an integrated report of ten chapters that bring together our research results that span from 2007 through this project year's effort. These chapters address: (1) Cost-Sensitive Acquisition of Open Architecture Software Systems; (2) Open Architectures for Software Systems; (3) License Challenges for Open Architectures; (4) Software License Legal Foundations; (5) Automating License Analysis; (6) Understanding the Role of Licenses and Evolution in Open Architecture Software Ecosystems; (7) Processes in Securing Open Architecture Software Systems; (8) Addressing Challenges in the Acquisition of Secure Software Systems with Open Architectures; (9) Ongoing Software Development without Classical Requirements; (10) Discussion and Recommendations. Specific recommendations that follow from our research that address the question, *How best to improve and streamline acquisition processes for secure OA systems,* can be identified as followed (and elaborated in Chapter 10, Volume II, this Report):

- *Encourage the adoption of acquisition business models in open source formats*
- *Encourage the development, (re)use and refinement of open source models of acquisition processes*
- *Develop and employ techniques for streamlining acquisition of secure OA systems, via*
    - *Acquisition process measurement and assessment*
    - *Acquisition process redesign and evolution*
    - *Design new acquisition processes*
    - *Cost management as an acquisition process design element*

These technical details, research integration, and more are found within Volume II of this Final Report. Last, it is our opinion that the compilation and integration of concepts, techniques, and materials presented in Volume II is a work in progress, and so it will benefit from ongoing refinement going forward, hopefully to be shown as part of our new (2015-16) acquisition research project now in progress.

Overall, we welcome any comments or questions on our research efforts, results, or recommendations.

References

[DoDOSA13] Department of Defense Open Systems Architecture Data Rights Team (2013). *DoD Open Systems Architecture Contract Guidebook for Program Managers* (Version 1.1). DoD, June 2013. https://acc.dau.mil/OSAGuidebook

[SEI13] Software Engineering Institute (2013). Managing Intellectual Property in the Acquisition of Software-Intensive System, November.

Table of Contents

Volume I: Achieving Better Buying Power through Acquisition of Open Architecture Software Systems for Web and Mobile Devices

Volume II: Understanding Open Architecture Software Systems: Licensing and Security Research and Recommendations

Chapter 1:

Research Overview

Introduction

This research focuses on continuing investigation and refinement of techniques for identifying and reducing the costs, streamlining the process, and improving the readiness of future workforce for the acquisition of complex software systems. Emphasis is directed at identifying, tracking, and analyzing software component costs and cost reduction opportunities within acquisition life cycle of open architecture (OA) systems, where such systems combine best-of-breed software components and software products lines (SPLs) that are subject to different intellectual property (IP) license requirements [DoDOSA13, SEI13].

This chapter provides an overview of the research effort during the period of 1 May 2014 through 31 May 2015. It includes a statement of work and description of the four research activities engaged during this period, followed by identification of the two acquisition research problems being investigated, the research and development basis for our research, and identification of our research publications that contain our studies and results. Each section is presented in turn.

Statement of Work and Research Description

Our objective was to develop new ways and means for identifying, tracking, and analyzing the costs associated with the acquisition life cycle of OA software systems. OA systems are those whose software elements can include either OSS or proprietary CSS components, where components are subject to different IP licenses and security constraints. Such components may be configured into different, functionally similar versions that allow for common but costly CSS components to be replaced by their OSS counterparts, as a strategy to reduce software acquisition costs. Such replacement or substitution may arise at different stages of system acquisition including system design, integration, deployment, and evolution. Recent DoD policy encourages the move to component-based OA software systems [DAU12, DISA12, DISA12a, DoDOSA11], especially as DoD moves to embrace new mobile computing devices like smartphones and cloud-based software application services [DISA12a, Tak12].

Better Buying Power (BBP, http://bbp.dau.mil/) is part of DoD's mandate to do more without more by implementing best practices in acquisition. BBP (up through 2013) identifies seven areas of focus organizing 36 initiatives that offer the potential to restore affordability in defense procurement and improve defense industry productivity. One area focuses on promoting competition, and includes an initiative to “enforce open system architectures and effectively manage technical data rights” [DAU12]. Technical data rights pertain to two categories of IP: the Government's rights to (a) technical data (TD – product design data, computer databases, computer software documentation, etc.); and (b) computer software (CS – source code, executable code, processes, and related materials). These rights are controlled by IP licenses from system product or service providers (i.e. software producers) to the Government customer, imposing obligations the customer must fulfill (e.g., a fee paid in exchange for a certain number of software users authorized for the licensed product or service) [An12]. Our acquisition research has focused on issues addressing OA systems and IP licenses since 2008 [ScA08].

OA software systems, integrated from components independently developed by different producers, offer the potential to reduce acquisition costs through new ways and means to acquire, develop, deploy, and sustain software-intensive systems. This may transform how DoD acquires complex systems by moving away from long-duration, proprietary (closed) system architectures with development costs that are difficult to control, towards more rapidly assembled/integrated OA systems with more transparent costs [ReB12, ScA13a, ScA13c]. Such a transformation may in turn reduce vendor lock-ins for deployed systems, often associated with rising costs and systems that are inaccessible to competing vendors. Our research on OA systems dating many years back [ScA08] has consistently been aligned with efforts to improve competition in software system development and evolution, through investigation of innovative ways and means to acquire/develop component-based OA software systems subject to diverse, heterogeneous IP licenses [AIS10]. But there is more to do to improve competition and defense affordability while effectively managing technical data rights in the acquisition of secure OA systems. There is a need to better understand how processes for acquiring cost-sensitive software systems are facilitated or constrained in light of overall BBP guidance and best practices, as well as how best to improve and streamline these processes when component-based OA software systems are being acquired.

We have sought to identify ways and means for streamlining the acquisition process for secure OA systems through new ways and means for identifying, tracking, and analyzing OA software component costs. Such systems often integrate components independently developed by different software producers as either OSS or proprietary CSS. Program managers, acquisition officers, and contract managers will increasingly be called on to review and approve security measures employed during the design, integration, deployment, and evolution of OA systems [DoDOSA11, ScA13b, ScA13c]. Our effort builds on both our prior acquisition research [e.g., ScA08, ScA11, ScA12b, ScA13a] and related acquisition research efforts at the PEO IWS [GuC10, GuW12, WoS11], Department of the Navy [MaS12], and Software Engineering Institute (SEI) that address SPLs [BeJ10, JoB11]. It is also influenced by related research in the DoD community addressing OSS [DISA12, HiW10, Ke12, MarL11], component-based software ecosystems [ReB12, ScA12b, ScA13c], and BBP initiatives [DAU12].

Realizing the objective of our investigation focused on *four project work activities*:

- Investigating the interactions between software system acquisition guidelines and processes, and the cost consequences of alternative software system architectures incorporating different mixes of OSS and CSS components subject to different licenses within secure OA SPLs [ScA08, ScA12b, ScA13a, ScA13b, ScA13c]. This entails exploring the balance between development, verification, and validation of software licenses and security rights, as well as the software component/license costs while managing the development and evolution of OA systems at design-time, build-time, and release and run-time.
- Developing formal foundations for establishing acquisition guidelines program managers can use in reduced cost acquisition of software-intensive systems that rely on development and deployment of secure OA systems using OSS and SPL technology and processes [AIS10, AIS13, ScA11, ScA12a, ScA12b, ScA13a, ScA13b, ScA13c].
- Continuing to develop concepts contributing to the emerging design of an automated approach supporting acquisition of secure OA systems by (a) determining their conformance to acquisition guidelines/policies, contracts, and related license management issues, and (b) giving future acquisition workforce support and insights to properly review, approve, and manage the acquisition of complex systems that incorporate cost-sensitive acquisition of OA systems and software components [AIS10, ScA11, ScA12a, ScA12b, ScA13a, ScA13b, ScA13c].
- Documenting the investigation, foundations, and results of the research in: (a) a technical Final Report delivered to the Technical Point of Contact at NPS; (b) a research presentation at the *11th Annual Acquisition Research Conference*, in Monterey, CA, May 2014; (c) a progress report with the OSD sponsor and others of interest within the OUSD (AT&L) offices; and (d) related research venues and publications, including periodic research progress reports.

Overall, we sought to identify, track, and analyze ways and means for how to articulate, tailor, and streamline the process for acquiring different kinds of secure OA systems. We seek to do so in ways that focus on software cost drivers and highlight opportunities for cost reduction through alternative software components or system configurations. Our investigation and research results are applicable to complex software elements used in many kinds of component-based OA software-intensive systems within DoD, as well as within other government agencies and industrial firms.

Scientific Background

The move to OA systems represents a transition from the acquisition of monolithic systems to the acquisition of reusable system components that can be integrated to realize different configurations of a software product line for a specific application domain [BeJ10, GuC10, JoB11, ReB12, ScA12b, WoS11]. These components are acquired within a software ecosystem that is evolving towards component provisioning within open repositories, where components from different producers are available for selection, evaluation, and system integration [GuW12, Iba13, MartL11, ReB12, ScA12a, ScA13b]. However, current scientific understanding of software system costs focus attention to estimating the cost of development for new proprietary CSS systems, that do not anticipate use of OSS, nor the replacement/substitution of CSS with functionally similar OSS components [MadB11].

OSS represents an integrated web of people, processes, and organizations, including project teams operating as virtual organizations. The “purchase price” for most OSS is “no cost” meaning it can be downloaded and used without additional software license fees, subject to compliance with the OSS component's license. Consequently, there is a basic need to understand how to identify an optimal mix of OSS and CSS components within OA systems, as well as how they reduce or increase the system costs during design, integration, deployment, and evolution when OSS components may be substituted for CSS components. However, the relationship among OA, OSS, security requirements, and acquisition is complex and evolving, so consequently it is poorly understood [cf. Sca09, Sca10, ScA11, ScA12b, ScA13c]. Subsequently, in 2007-08, we began by examining how different OSS licenses can encumber software systems with OA, which give rise to new requirements for how best to acquire OA software-intensive systems the employ OSS software elements [ScA08] during system design, integration, deployment, and evolution [ScA13a, ScA13b].

Our recent acquisition research efforts demonstrate it is both possible and feasible to develop OA systems that incorporate best-of-breed software components, whether proprietary CSS or OSS, in ways that can reduce the initial and sustaining acquisition costs of such systems.

We strongly believe that our research results are applicable to enterprise information systems, which are widespread throughout DoD and the U.S. government, as well as to command and control (C2) systems (e.g., [ReB12, ScB12, ScA13b]) and other defense systems. The audiences for our presentations included in Chapters 2, 3, and 4 in this Final Report Volume I, and the compiled and integrated materials we are developing, as included in Volume II of this Final Report, demonstrate our commitment to communicate our research results to large, diverse audiences. Doing so however requires new guidance, and ideally automated tools, for explicitly modeling and analyzing the architecture of an OA system during its development and evolution, along with annotating the architecture with software component license rights and obligations. Our results thus demonstrate a major technological advance in the acquisition and development of OA systems, as a breakthrough in simplifying software license analysis throughout the contracting activities. Creating similar advances for streamlining the acquisition process while reducing the costs of secure OA systems is the next breakthrough that is needed.

Acquisition Research Problems and Our Approach:

The core of our proposed technical approach was to investigate a closely related set of research questions through systematic empirical observation of current software cost and IP licensing practices for different kinds of common application and infrastructure software components. We then sought to formalize and comparatively analyze these observations and practices into computational schemes that supplement our existing framework for modeling and analyzing the licensing and security requirements of OA software systems. In short, we sought to extend our formal software IP modeling scheme to incorporate software component cost elements, in ways derived from the answers to our four research questions. These four research questions follow from our accomplishments described above and in detail elsewhere [ScA13a, ScA13b, ScA13c]. Each is described in turn.

Research question 1: ***How best to identify, track, and analyze OA-driven software design, integration, deployment, and evolution costs.*** Researchers and practitioners have identified various kinds of software component/system design, integration, deployment, and evolution costs: one-time purchase of license rights, subscription-like licensing, usage-based licensing; source code licensing, technical data licensing, service licensing, licensing of data generated while the system runs; licensing of software elements needed at run time, deployment time, distribution time, build time, design time; training and support needed for a system to be usable; evolution costs to maintain the current system; evolution costs to reach future versions; evolution costs to branch out into related systems to enable reuse; and so forth. Which of these costs matters to whom, when, where, and why, and which should people in the acquisition workforce be expected to track and manage?

Research question 2: ***How best to identify, track and analyze how OA software system integration, deployment, and evolution costs are linked to OA design decisions.*** Software system architectural decisions influence overall costs, but in what way? A brute-force approach could estimate overall system life cycle costs for each candidate architecture. Guidance for an architectural decision then requires calculating this estimate for each alternative that the decision may produce. The question of whether more direct connections may be made from specific classes of costs to specific kinds of decisions is an open one. How are people in the acquisition workforce to make such decisions and realize predictable costs of alternative system architectures?

It is clear that overall system life cycle costs cannot in general be evaluated without information about the context in which the system is expected to be developed, built, distributed, deployed, used and evolved. The effect of architectural decisions on overall acquisition life cycle costs necessarily takes place and is strongly influenced by the context in which the costs are incurred, and an important role is played by such questions as how many instances of the system are expected, how much and what kind of usage is expected, over what time period, with what level of training and support, with what expected future evolution, preferring which suppliers and ecosystems, and so forth. It will be necessary to characterize the kinds of context that are needed, and to express contexts in a way that is manageable for the people who must collect or estimate the information as well as for the architects, acquisition officers, and others that will use it in making decisions that will affect costs. We foresee that choosing among the contexts will be in some sense as influential and important as choosing among the myriad architectural alternatives. In any case, making sensible architectural choices will not be practical without appropriate characterizations of the contexts in which their effects will unfold.

Research question 3: ***How best to analyze possibly-incommensurable OA system component costs, system requirements, and IP obligations.*** In our previous work we have focused on non-monetary license obligations. While some obligations appear in more than one license, and a few appear to be very widely distributed across licenses, in general each license requires its own distinct list of obligations in exchange for the rights offered. An important result from our research has been an approach for placing license obligations (and rights as well) in a partial order, based on subsumption among the classes of actions that satisfy the obligations. Using this we can show that one license obligation subsumes another, or stated informally that satisfying the first obligation necessarily satisfies the second obligation as well. The subsumption relationship among obligations makes reasoning about licenses a manageable task. The monetary obligations of various kinds that proprietary licenses impose must be brought into this partial order.

While costs of the same class are ordered based on the numeric value involved, there are (as we note above) different classes of costs such as purchase costs, subscription costs, per-seat costs, support costs, and so forth. We see several avenues that appear promising, such as (to list a few of the more obvious ones) comparing an outright purchase cost with a subscription cost summed over the expected lifetime of the system, or subscription costs over different periods by converting them to the costs for a specific time period of interest. The many classes of costs raise the need for approaches for comparing or if possible unifying them. In addition, costs and other non-monetary license obligations must be considered together in a useful fashion.

Research question 4: ***How best to present OA system design guidance for identify, tracking, and analyzing software integration, deployment, and continuous evolution costs.*** It is essential to collect and calculate information about the effect of architectural decisions on overall costs, but just as essential to be able to combine and present it in a way that provides usable architectural guidance. Research questions in this phase include identifying relevant architectural decisions, marshalling the cost information relevant to each decision, and evaluating alternatives in a way that allows architects to make good choices. While a total of overall system costs, including monetary costs and non-monetary obligations that must be met, is a fundamental and important criterion, we foresee that more focused information will also be useful. For example, certain kinds of architectural decisions affect the evolution of the system's software ecosystem in ways that may not directly translate into costs but still have a powerful effect on whether the system will thrive in the future, such a steering the system away from closed interfaces and proprietary solutions toward open interfaces and solutions that can available from a variety of suppliers.

Overall, investigating and developing answers to these four questions is the focus of our proposed effort for the 2013-2014 project period. In particular, we seek to develop, document, and deliver our answers through research publications that will be presented at the *2014 Acquisition Research Symposium* and elsewhere. We similarly seek to articulate these answers in ways that can ultimately contribute to the practice and guidance provided to the acquisition workforce.

Inter-project research coordination

We continue to believe we are and have been extremely well positioned to leverage our recent research work and results [AlS10, ScA08, ScA11, ScA12a, ScA12b, ScA13a, ScA13b, ScA13c] with the effort described in this Final Report. We have continued to build on our current research efforts in OSS [e.g., Sca10] and software requirements-architecture interactions [ScA08, Sca09], as well as our track record in prior acquisition research studies. Similarly, we have found recent related research supported by the DoD addressing related issues in OSS [HiW10] also influences our proposed effort. In addition, our effort builds from and contributes to research on software system acquisition within the DoD, focusing on software reuse [MaS12], SPLs [GuW10, BeJ10], open innovation and emerging software component markets [GuW12]. We thus believe our complementary research places us at an extraordinary advantage to conduct the proposed study that addresses a major strategic acquisition goal of the DoD and the military services [DoDOSA11].

Prospects for longer-term Acquisition-related research

The military services have committed to orienting their major system acquisition programs around the adoption of an OA strategy that in turn embraces and encourages the adoption, development, use, and evolution of OSS [DoDBBP14, DoDGSA13, DoDOSA11]. Thus, there is a significant need for sustained research that investigates the interplay and inter-relationships between:

- (a) current/emerging guidelines for the acquisition of software-intensive systems within the DoD community (including contract management and software development issues);
- (b) how secure, reusable software product lines [MaS12, WoS11] that employ an OA incorporating OSS/CSS component products (e.g., widgets, apps, and mashups) and their production processes [ScA13b], are essential to improving the BBP and cost-reduction effectiveness of software-intensive program acquisition efforts; and
- (c) how (a) and (b) contribute to advances and new insights for how best to realize the Better Buying Power initiatives addressing open architecture systems that may incorporate open source software components and closed source software components that are subject to different, possibly conflicting Intellectual Property (IP) licenses and cybersecurity requirements.

Research Results

Our research studies and results are included in the remaining chapters of this Final Report as three research presentations. These presentations served to document our effort to disseminate our research results from this research project to different communities of interest, within the DoD and the Federal Government. Versions of these presentations and related materials have also been circulated within the Assembled Capabilities Working Group (ACWG) and the Defense Intelligence Information Enterprise (DI2E). The ACWG is an effort directed by the Command, Control, Communications, Cyber and Business (C3CB) office, within the OUSD (AT&L), and facilitated by The MITRE Corporation. There are approximately 300 participants within the ACWG at this time. Further, information requests from participants within the ACWG, DI2E, mission partners, and other parts of the Federal Government gave rise to the need to produce an integrated set of research results and recommendations from our effort. This in turn gave rise to our effort to produce the ten chapter, Volume II of this Final Report. Thus, our overall research results are described and documents within this two volume Final Report.

Summary presentations that have been publicly shared resulting from our research appear in Chapter 2, 3, and 4 of this Report in Volume I. Chapter 2 includes the abstract and slide deck that were presented at the 2014 Acquisition Research Symposium (May 2014). Chapter 3 is the slide deck from MITRE-ATARC Workshop in Washington, DC (August 2014) addressing Cost-Sensitive Acquisition of Open Architecture Software Systems for Mobile Devices. Chapter 4 is the slide deck from the Federal Mobile Computing Summit also held in Washington, DC (August 2014).

In Volume II of this Report, we have compiled an integrated report of ten chapters that bring together our research results that span from 2007 through this project year's effort. These chapters address: (1) Cost-Sensitive Acquisition of Open Architecture Software Systems; (2) Open Architectures for Software Systems; (3) License Challenges for Open Architectures; (4) Software License Legal Foundations; (5) Automating License Analysis; (6) Understanding the Role of Licenses and Evolution in Open Architecture Software Ecosystems; (7) Processes in Securing Open Architecture Software Systems; (8) Addressing Challenges in the Acquisition of Secure Software Systems with Open Architectures; (9) Ongoing Software Development without Classical Requirements; (10) Discussion and Recommendations. Specific recommendations that follow from our research that address the question, *How best to improve and streamline acquisition processes for secure OA systems,* can be identified as followed (and elaborated in Chapter 10, Volume II, this Report):

- *Encourage the adoption of acquisition business models in open source formats*
- *Encourage the development, (re)use and refinement of open source models of acquisition processes*
- *Develop and employ techniques for streamlining acquisition of secure OA systems, via*
    - *Acquisition process measurement and assessment*
    - *Acquisition process redesign and evolution*
    - *Design new acquisition processes*
    - *Cost management as an acquisition process design element*

These technical details, research integration, and more are found within Volume II of this Final Report. Last, it is our opinion that the compilation and integration of concepts, techniques, and materials presented in Volume II is a work in progress, and so it will benefit from ongoing refinement going forward, hopefully to be shown as part of our new (2015-16) acquisition research project now in progress.

Overall, we are grateful for the support and funding we have received that enabled our acquisition research to continue and be documented, as shown in this Final Report.

Acknowledgements

Preparation of this report and all work products therein benefitted from a research grant, #N00244-14-1-0030 from the Acquisition Research Program at the Naval Postgraduate School, Monterey, CA. None of the content of this Final Report has been reviewed, approved, or endorsed by the ARP, NPS, US Navy, Department of Defense or any other government agency. The work presented is solely the responsibility of the authors.

References

[AlS10] Alspaugh, T.A, Scacchi, W., and Asuncion, H. (2010). Software Licenses in Context: The Challenge of Heterogeneously Licensed Systems, *J. Assoc. Information Systems*, 11(11), 730-755, November 2010.

[AlS13] Alspaugh, T.A. And Scacchi, W. (2013). Ongoing Software Development without Classical Requirements, *Proc. 21st Intern. Conf. Requirements Engineering*, Rio de Janeiro, BZ, 165-174, July 2013.

[An12] Anderson, S. (2012). Software Licensing – Smart Spending in These Changing Times, *CHIPS: The Department of the Navy's Information Technology Magazine*, July, 28-31.

[BeJ10] Bergey, J., & Jones, L. (2010). Exploring acquisition strategies for adopting a software product line approach. *Proc. 7th Acquisition Research Symposium*. Vol. 1, 111-122, Naval Postgraduate School, Monterey, CA.

[DAU12] Defense Acquisition University (2012). *Open Systems Architecture and Technical Data Rights...Management Approaches*, http://1.usa.gov/Zl6dRT accessed 30 October 2012.

[DISA12] Defense Information Systems Agency (2012). *DOD Open Source and Community Source Software Development in Forge.mil*, SoftwareForge Document ID – doc26066doc26066, http://bit.ly/16abVh1, accessed 30 October 2012

[DISA12a] Defense Information Systems Agency (2012). *Strategic Plan: 2013-2018*, Version 1.0, http://www.disa.mil/News/PressResources/2012/~/media/Files/DISA/About/Strategic-Plan.pdf, accessed 30 October 2012.

[DoDBBP14] Department of Defense, *Better Buying Power* http://bbp.dau.mil/ accessed 2014.

[DoDGSA13] Department of Defense and General Services Administration (2013). *Improving Cybersecurity and Resilience through Acquisition*, November 2013, accessed June 2014. http://www.defense.gov/news/Improving-Cybersecurity-and-Resilience-Through-Acquisition.pdf

[DoDOSA11] Department of Defense Open Systems Architecture (2011). *Contract Guidebook for Program Managers*, Vol. 0.1, December, https://acc.dau.mil/OSAGuidebook

[DoDOSA13] Department of Defense Open Systems Architecture Data Rights Team (2013). *DoD Open Systems Architecture Contract Guidebook for Program Managers* (Version 1.1). DoD, June 2013. https://acc.dau.mil/OSAGuidebook

[Gi11] Gizzi, N. (2011). Command and Control Rapid Prototyping Continuum (C2RPC) Transition: Bridging the Valley of Death, *Proc. 8th Annual Acquisition Research Symposium*, Vol. 1, 135-154, Naval Postgraduate School, Monterey, CA.

[GuC10] Guertin, N. and Clements, P. (2010). Comparing Acquisition Strategies: Open Architecture versus Product Lines, Vol. 1, 78-90, *Proc. 7th Acquisition Research Symposium*, Naval Postgraduate School, Monterey, CA.

[GuW12] Guertin, N. and Womble, B. (2012). Competition and the DoD Marketplace, *Proc. 9th Acquisition Research Symposium*. Vol. 1, 76-82, Naval Postgraduate School, Monterey, CA.

[HiW10] Hissam, S., Weinstock, C.B., and Bass, L. (2010). On Open and Collaborative Software Development in the DoD, Vol. 1, 219-235, *Proc. 7th Acquisition Research Symposium*, Naval Postgraduate School, Monterey, CA.

[Iba13] Ibanez, L. (2013). Ozone Widget Framework required to be open source under congressional law, 5 March 2013, http://opensource.com/education/13/2/ozone-widget-framework , accessed 8 June 2013. Also see, *Ozone Widget Framework*, https://www.owfgoss.org/

[JoB11] Jones, L. and Bergey, J. (2011). An Architecture-Centric Approach for Acquiring Software-Reliant Systems, *Proc. 8th Acquisition Research Symposium*, Vol. 1, 32-49, NavalPostgraduate School, Monterey, CA.

[Ke12] Kenyon, H. (2012). DoD, Intel Officials Bullish On Open Source Software; Government-wide Software Foundation In The Mix, *AOL Defense,* October 2012.

[MaS12] Mactal, R., Spruill, N. (2012). A Framework for Reuse in the DoN. *Proc. 9th Acquisition Research Symposium,* Vol.1, 149-164, Naval Postgraduate School, Monterey, CA.

[MadB11] Madachy, R, Boehm. B., Clark, B., Tan, T., and Rosa, W. (2011). US DoD Application Domain Empirical Software Cost Analysis, *2011 Intern. Symp. Empirical Software Engineering and Measurement*, Banff, Canada, 392-395.

[MarL11] Martin, G. and Lippold, A. (2011). Forge.mil: A Case Study for Utilizing Open Source Software Inside of Government, *Open Source Systems*, Springer, 334-337.

[MatH12] Mathieu, J., Fulk, M., Lorber, Klein, G., Costam B. and Schmorrow, D. (2012). Social Radar Workflows, Dashboards, and Environments, MITRE Corporation, RTO-MP-HFM-201, Paper 25-1, http://www.mitre.org/work/tech_papers/2012/12_0567/12_0567.pdf

[ReB12] Reed, H., Benito, P., Collens, J., and Stein, F. (2012). Supporting Agile C2 with an Agile and Adaptive IT Ecosystem, *17th. Intern. Command and Control Research and Technology Symposium* (ICCRTS), Paper-044, Fairfax, VA, June 2012.

[Sca09] Scacchi, W., (2009). Understanding Requirements for Open Source Software, in K. Lyytinen, P. Loucopoulos, J. Mylopoulos, and W. Robinson (eds.), *Design Requirements Engineering: A TenYear Perspective*, LNBIP 14, Springer Verlag, 467-494, 2009.

[Sca10] Scacchi, W. (2010). The Future of Research in Free/Open Source Software Development, *Proc. ACM Workshop Future of Software Engineering Research* (FoSER), Santa Fe, NM, 315-319.

[ScA08] Scacchi, W. and Alspaugh, T., (2008). Emerging Issues in the Acquisition of Open Source Software within the U.S. Department of Defense, *Proc. 5th Acquisition Research Symposium*, NPS-AM-08-036, Naval Postgraduate School, Monterey, CA, May.

[ScA11] Scacchi, W. and Alspaugh, T., (2011). Advances in the Acquisition of Secure Systems Based on Open Architectures, *Proc. 8th Acquisition Research Symposium*, Vol. 1, Naval Postgraduate School, Monterey, CA.

[ScA12a] Scacchi, W. and Alspaugh, T., (2012a) Understanding the Role of Licenses and Evolution in Open Architecture Software Ecosystems, *J. Systems and Software*, 85(7), 1479-1494, July 2012.

[ScA12b] Scacchi, W. and Alspaugh, T., (2012b). Addressing Challenges in the Acquisition of Secure Software Systems with Open Architectures, *Proc. 9th Acquisition Research Symposium*, Vol. 1, 165-184, Naval Postgraduate School, Monterey, CA.

[ScA13a] Scacchi, W. and Alspaugh, T., (2013a). Streamlining the Process of Acquiring Secure Open Architecture Software Systems, *Proc 10th Annual Acquisition Research Symposium,* Monterey, CA, 608-623, May 2013.

[ScA13b] Scacchi, W. and Alspaugh, T. (2013b). Processes in Securing Open Architecture Software Systems, *Proc. 2013 Intern. Conf. Software and System Processes*, 126-135, San Francisco, CA, May 2013.

[ScA13c] Scacchi, W. and Alspaugh, T. (2013c). Challenges in the Development and Evolution of Secure Open Architecture Command and Control Systems, *Proc. 18Th Intern. Command and Control Research and Technology Symposium*, Paper-098, Alexandria, VA, June 2013.

[ScA14a] Scacchi, W. and Alspaugh, T. (2014). Achieving Better Buying Power through Cost-Sensitive Acquisition of Open Architecture Software Systems. *Proc 11th Annual Acquisition Research Symposium*, Monterey, CA, NPS-AM-14-C11P07R01-036, May 2014.

[ScA14b] Scacchi, W. and Alspaugh, T. (2014). Cost-Sensitive Acquisition of Open Architecture Software Systems for Mobile Devices, Invited Presentation, MITRE-ATARC Workshop on Challenges in Legal and Acquisition, *Federal Mobile Computing Summit*, Washington, DC, 19 August 2014.

[ScA14c] Scacchi, W. and Alspaugh, T. (2014). Reasoning about the Security of Open Architecture Software Systems for Mobile Devices, Invited Presentation, *Federal Mobile Computing Summit*, Washington, DC, 20 August 2014.

[ScB12] Scacchi, W., Brown, C. and Nies, K. (2012). Exploring the Potential of Virtual Worlds for Decentralized Command and Control, *Proc. 17th. Intern. Command and Control Research and Technology Symposium* (ICCRTS), Paper 096, Fairfax, VA, June 2012.

[SEI13] Software Engineering Institute (2013). *Managing Intellectual Property in the Acquisition of Software-Intensive System*, November.

[Tak12] Takai, T.M. (2012). *Department of Defense Mobile Device Strategy*, Version 2.0, Office of the DoD Chief Information Officer, May 2012. http://www.defense.gov/news/dodmobilitystrategy.pdf accessed May 2013.

[WoS11] Womble, B., Schmidt, W., Arendt, M., and Fain, T. (2011). Delivering Savings with Open Architecture and Product Lines, *Proc. 8th Acquisition Research Symposium*, Vol. 1, 8-13, Naval Postgraduate School, Monterey, CA.

Chapter 2:

Achieving Better Buying Power through Cost-Sensitive Acquisition of Open Architecture Software Systems

Achieving Better Buying Power through Cost-Sensitive Acquisition of Open Architecture Software Systems

Proposal for Presentation at the 11th Annual Acquisition Research Symposium

Monterey, CA

Walt Scacchi and Thomas A. Alspaugh

Institute for Software Research

University of California, Irvine

Irvine, CA 92697-3455 USA

Abstract

Our proposed presentation for the 11th Annual Acquisition Research Symposium in May 2014 focuses on our ongoing investigation and refinement of techniques for identifying and reducing the costs, streamlining the process, and improving the readiness of future workforce for the acquisition of complex software systems. Emphasis is directed at identifying, tracking, and analyzing software component costs and cost reduction opportunities within acquisition life cycle of open architecture (OA) systems, where such systems combine best-of-breed software components and software products lines (SPLs) that are subject to different intellectual property (IP) license requirements.

Research Issue

The Department of Defense, other government agencies, and most large-scale business enterprises continually seek new ways to improve the functional capabilities of their software-intensive systems. The acquisition of OA systems that can adapt and evolve through replacement of functionally similar software components is an innovation that can lead to lower cost systems with more powerful functional capabilities. Our research identifies and analyzes how software component costs and IP license requirements interact to drive down (or drive up) total system costs across the system acquisition life cycle. The availability of such new scientific knowledge and technological practices can give rise to more effective expenditures of public funds and improve the effectiveness of future software-intensive systems used in government and industry. Thus, a goal of this presentation is to support and advance a public purpose through acquisition research and results.

Research Result

Our research results identify a new approach to address Better Buying Power challenges in the acquisition of software systems for the Department of Defense. Program managers, acquisition officers, and contract managers will increasingly be called on to review and approve choices between functionally similar low or no cost open source software components, and commercially priced closed source software components, to be used in the design, implementation, deployment, and evolution of open architecture (OA) systems. We seek to make this a simpler, more transparent, and more tractable process. Such a process must identify, track, and analyze software component costs throughout the system life cycle, and be easy to reuse for different system application domains, in order to realize cost reductions and improve acquisition workforce capabilities. Our recent research demonstrates how complex OA systems can be designed, built, and deployed with alternative components and connectors resulting in functionally similar system versions, to satisfy overall system security requirements and individual system component intellectual property (IP) requirements. Our next step, to be presented here, is to identify, track, and analyze software component costs associated with different types of component IP licenses when acquiring OA systems, and to do so in ways that highlight opportunities for cost reduction. We believe our results are applicable to enterprise software systems in other government agencies and industrial firms, as well as to enterprise and mission-critical systems for the DoD community.

Presentation materials associated with this proposal follow on the next pages.

Achieving Better Buying Power through Cost-Sensitive Acquisition of Open Architecture Software Systems

Walt Scacchi and Thomas Alspaugh
Institute for Software Research
University of California, Irvine
Irvine, CA 92697-3455 USA

INSTITUTE *for* SOFTWARE RESEARCH
UNIVERSITY *of* CALIFORNIA • IRVINE

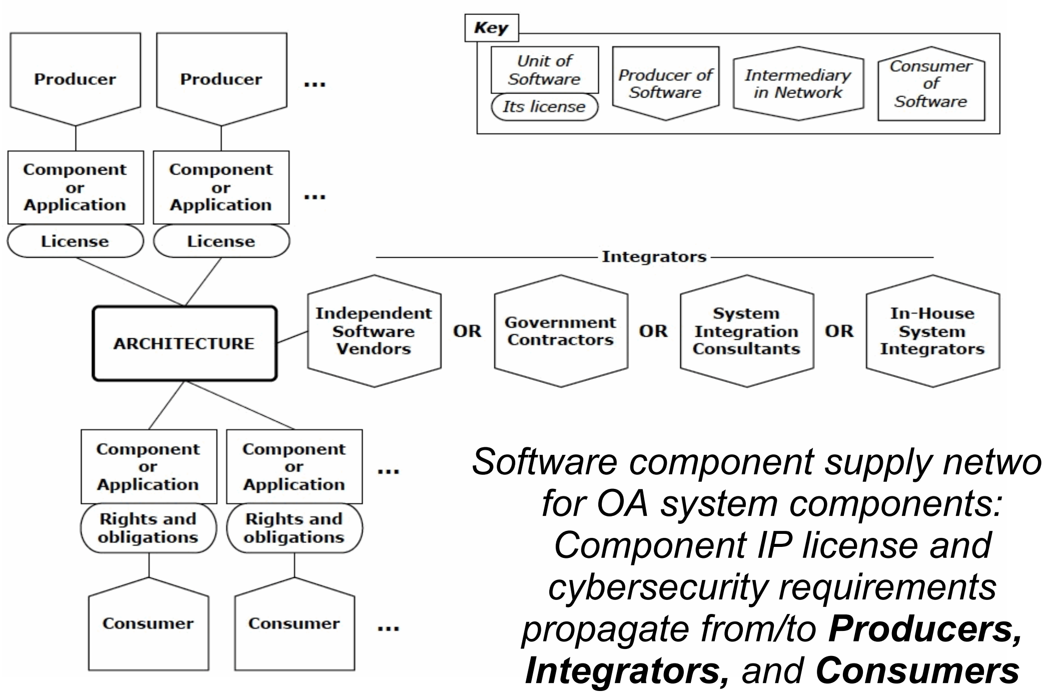
Overview

- Recent trends in open architecture (OA) software systems
- Emerging challenges in achieving *Better Buying Power* (BBP) via OA software systems
- New practices to realize cost-effective acquisition of OA software systems
- Conclusions

Recent trends in OA software systems

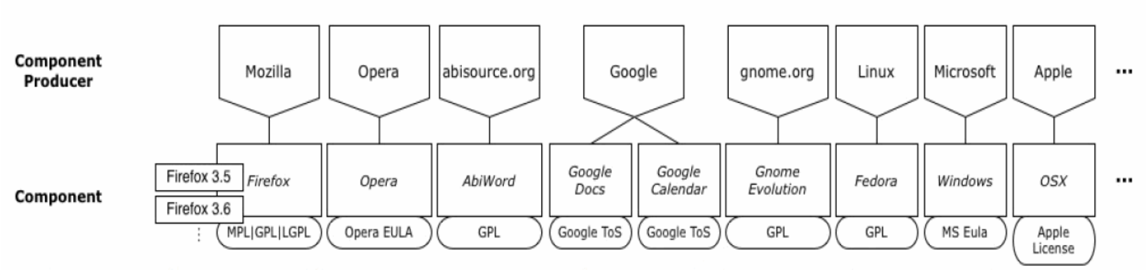
- Multi-party acquisition and OA development ecosystems
- Shared development of Apps and Widgets as OA system components
- Growing diversity of challenges in cybersecurity
- New business models for OA software component development and use

Multi-party acquisition and OA development ecosystems



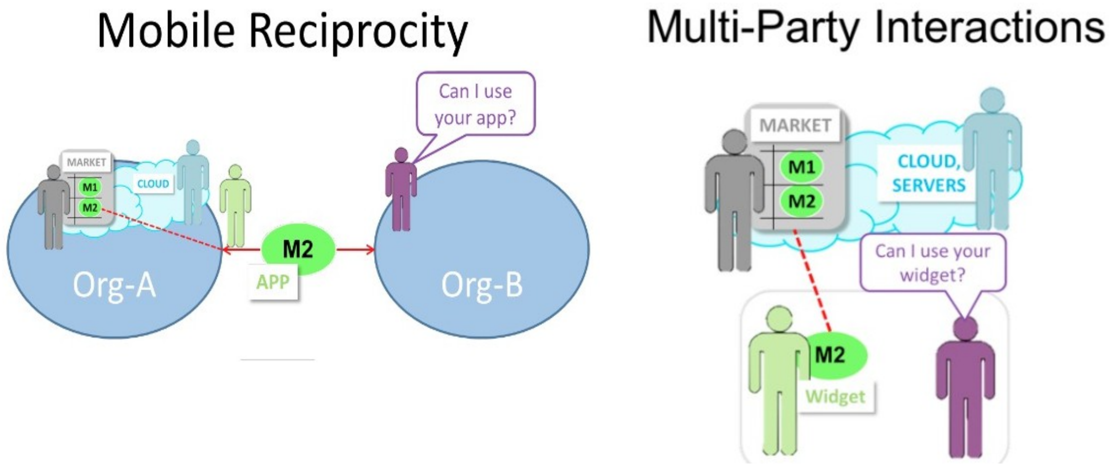
*Software component supply network for OA system components: Component IP license and cybersecurity requirements propagate from/to **Producers, Integrators,** and **Consumers***

Multi-party acquisition and OA development ecosystems



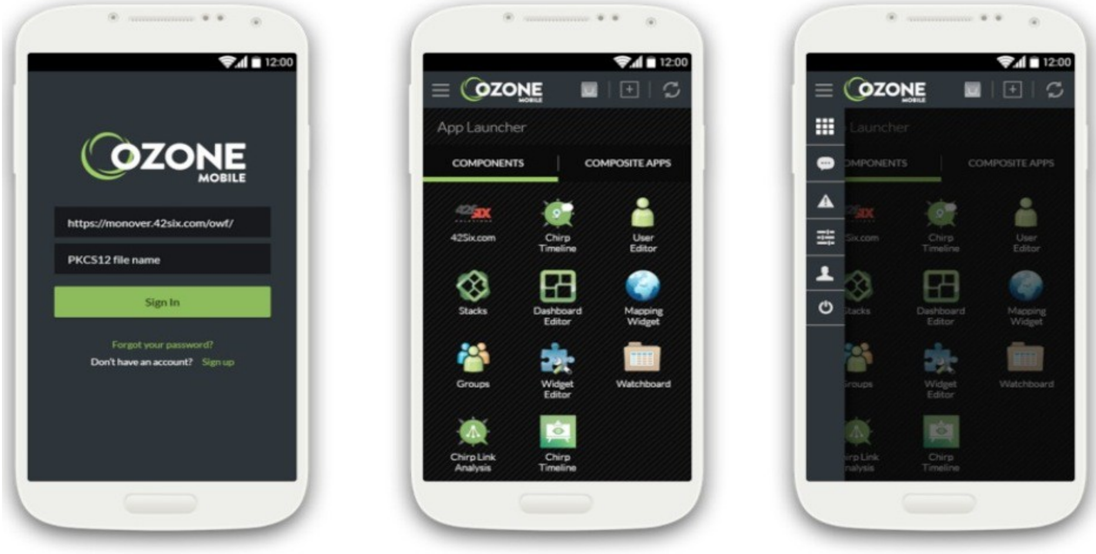
A sample elaboration of producers (vendors), software component applications, and IP licenses for OA system components.

Multi-party acquisition and OA development ecosystems



Consumer/End-User Organizations now looking for ways to reduce acquisition cost and effort through shared development/use of common OA software system components (apps, widgets).

Shared development of Apps and Widgets as OA system components

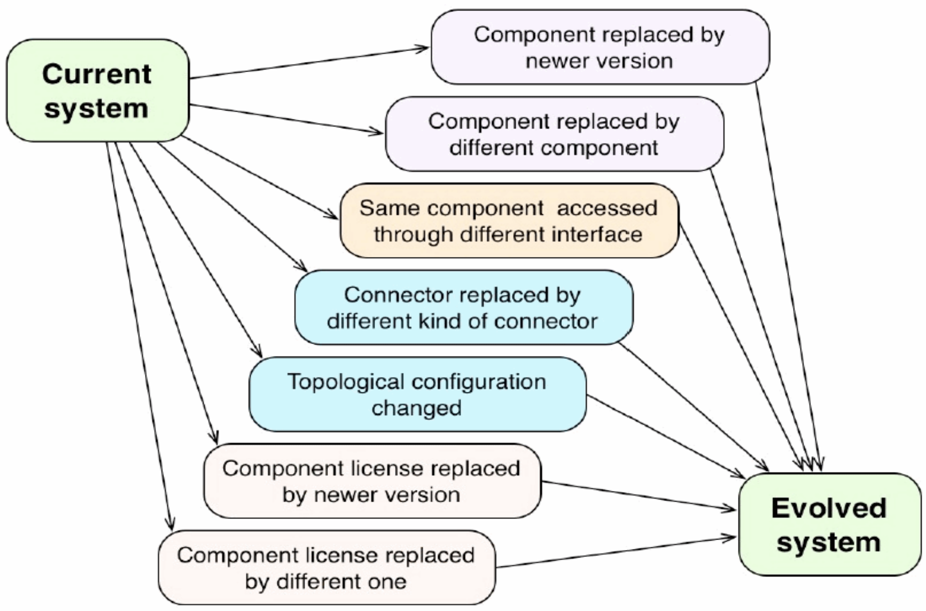

CAS Sign In

App Launcher

Ozone Mobile Drawer Menu

Ozone Widget Framework for Web PCs and Mobile Devices

Multi-party acquisition and OA development ecosystems: *Multiple OA system evolution paths*


Current system

Component replaced by newer version

Component replaced by different component

Same component accessed through different interface

Connector replaced by different kind of connector

Topological configuration changed

Component license replaced by newer version

Component license replaced by different one

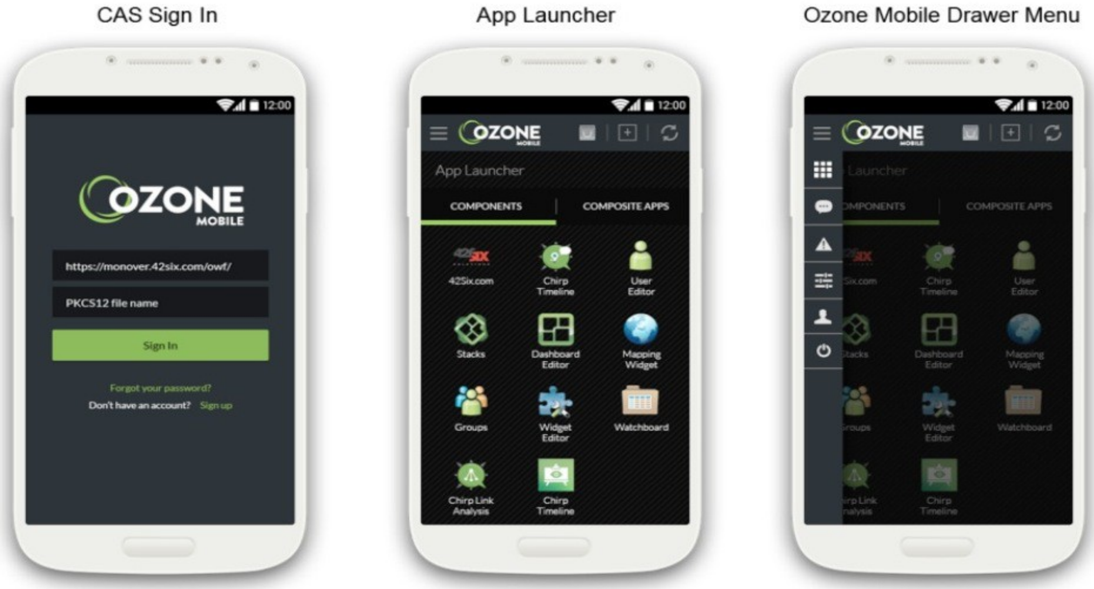
Evolved system

IP and cybersecurity requirements will need continuous attention!

Growing diversity of challenges in cybersecurity

- Scacchi, W. and Alspaugh, T. (2012) Addressing Challenges in the Acquisition of Secure Software Systems with Open Architectures, *Proc. 9th Acquisition Research Symposium*, Vol. 1, 165-184, Naval Postgraduate School, Monterey, CA.
- Scacchi, W. and Alspaugh, T. (2013a). Processes in Securing Open Architecture Software Systems, *Proc. 2013 Intern. Conf. Software and System Processes*, San Francisco, CA, May 2013.
- Scacchi, W. and Alspaugh, T.A. (2013b). Streamlining the Process of Acquiring Secure Open Architecture Software Systems, *Proc. 10th Annual Acquisition Research Symposium*, Monterey, CA, 608-623, May 2013.
- Scacchi, W. and Alspaugh, T.A. (2013c). Challenges in the Development and Evolution of Secure Open Architecture Command and Control Systems, *Proc. 18th Intern. Command and Control Research and Technology Symposium*, Paper-098, Alexandria, VA, June 2013.

Shared development of Apps and Widgets as OA system components: *Cybersecurity?*



Ozone Widgets supporting "Bring Your Own Devices" (BYOD)?

New business models for OA software components

- Franchising
- Enterprise licensing
- Metered usage
- Advertising supported
- Subscription
- Free component, paid service fees

- Federated reciprocity for shared development
- Collaborative buying
- Donation
- Sponsorship
- (Government) open source software
- and others

Managing acquisition costs will be demanding. Acquisition workforce will need automated assistance*, else acquisition management costs will dominate development costs for OA software components!*

Emerging challenges in achieving BBP via OA software systems

- Acquisition program managers/staff *may not understand* how software IP licenses affect OA system design, and vice-versa.
- Software IP and cybersecurity obligations and rights propagate across system development, deployment, and evolution activities *in ways not well understood* by system developers, integrators, end-users, or acquisition managers.
- *Failure to understand* software IP and cybersecurity obligations and rights propagation can reduce DoD buying power, increase software life cycle costs, and reduce competition.
- DoD and other Government agencies *would financially and administratively benefit* from engaging the development and deployment of an (open source) automated software obligations and rights management system for the acquisition workforce.

New practices to realize cost-effective acquisition of OA software systems

- Need to R&D ***worked examples*** of reference OA system models, and component evolution alternatives.
- Need ***open source models of*** app/widget security assurance ***processes and*** reusable cybersecurity ***requirements.***
- Need precise ***domain-specific languages*** (DSLs) and ***automated analysis tools*** for continuously assessing and continuously improving cybersecurity and IP requirements for OA C2 systems composed from apps/widgets.

Conclusions

- Our research identifies how new software component technologies, IP and security requirements, and new business models interact to drive-down or drive-up acquisition costs.
- New technical risks for component-based OA software systems can dilute the cost-effectiveness of BBP efforts.
- Need R&D leading to automated systems that can model and analyze OA system IP licenses and cybersecurity requirements
  - Empower the acquisition workforce
  - Identify and manage cost-effectiveness trade-offs

Acknowledgements

This material is based upon work supported by the Naval Postgraduate School Acquisition Research Program under Grant No. N00244-14-1-0030. The views expressed in materials or publications, and/or made by the presenters, do not necessarily reflect the official policies of the Naval Postgraduate School, nor does mentions of trade names, commercial practices, or organizations imply endorsement by the U.S. Government.

INSTITUTE *for* SOFTWARE RESEARCH
UNIVERSITY *of* CALIFORNIA • IRVINE

Chapter 3:

Cost-Sensitive Acquisition of Open Architecture Software Systems for Mobile Devices

Cost-Sensitive Acquisition of Open Architecture Software Systems for Mobile Devices

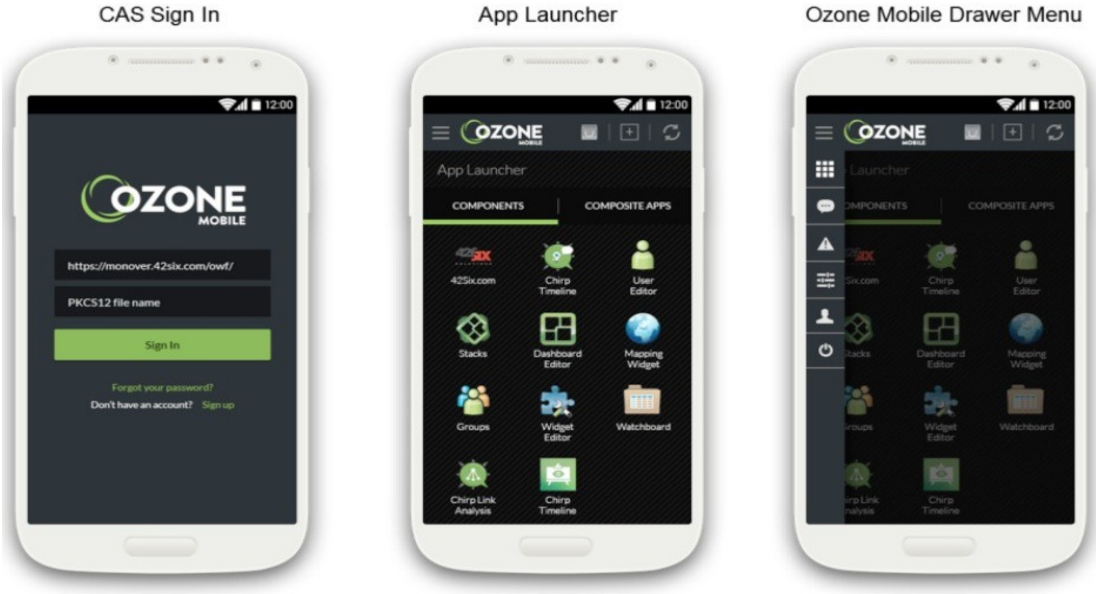
Walt Scacchi and Thomas Alspaugh
Institute for Software Research
University of California, Irvine
Irvine, CA 92697-3455 USA

INSTITUTE *for* SOFTWARE RESEARCH
UNIVERSITY *of* CALIFORNIA • IRVINE

Overview

- Recent trends in open architecture (OA) software systems
- Emerging challenges in achieving *Better Buying Power* (BBP) via OA software systems
- New practices to realize cost-effective acquisition of OA software systems
- Conclusions

Apps and Widgets as Mobile OA system components

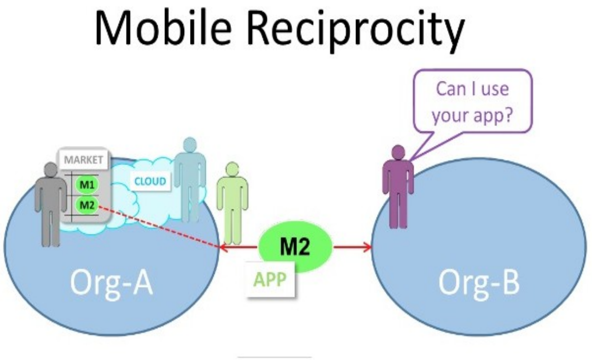

CAS Sign In

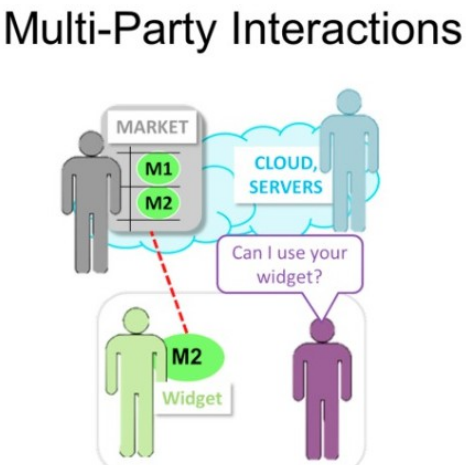
App Launcher

Ozone Mobile Drawer Menu

Ozone Widget Framework for Web PCs and Mobile Devices

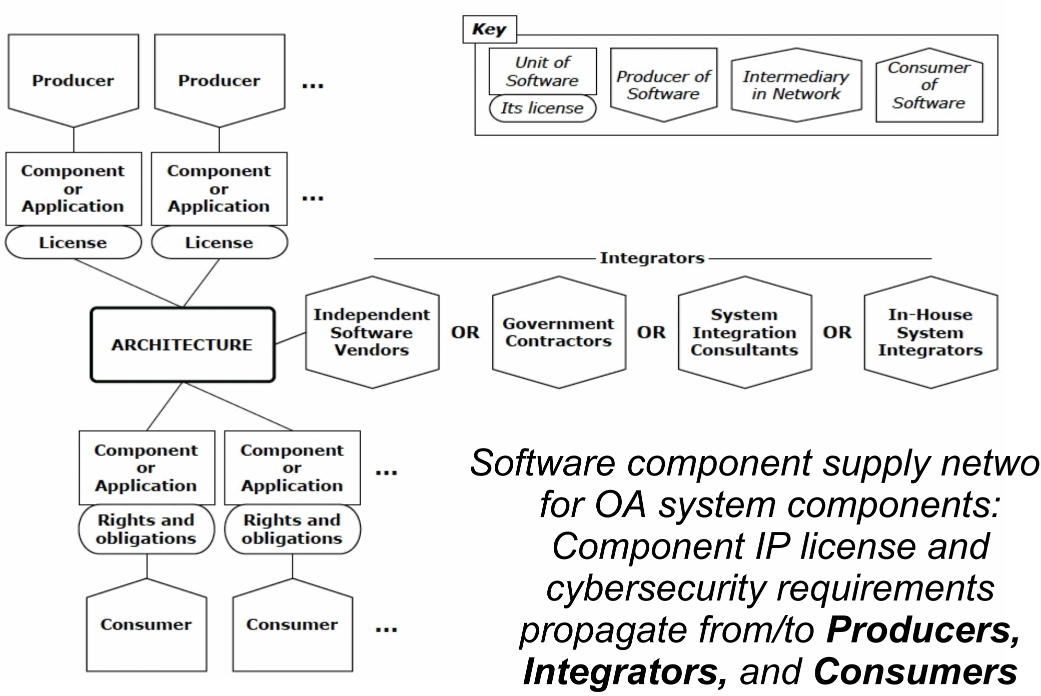
Multi-party acquisition and OA development ecosystems


Mobile Reciprocity


Multi-Party Interactions

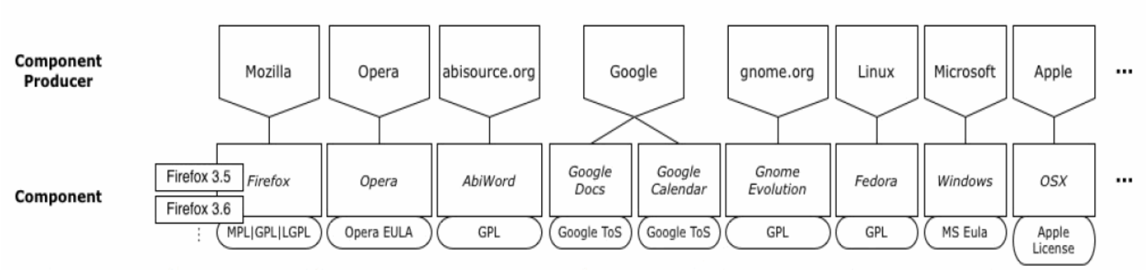
Consumer/End-User Organizations now looking for ways to reduce acquisition cost and effort through shared development/use of common OA software system components (apps, widgets).

Multi-party acquisition and OA development ecosystems



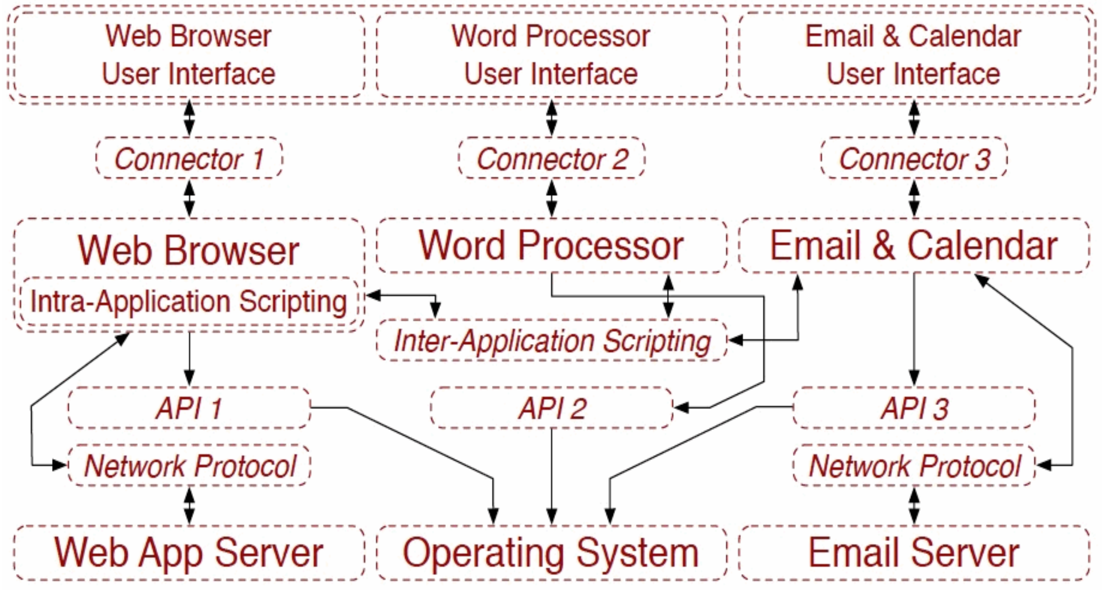
Software component supply network for OA system components: Component IP license and cybersecurity requirements propagate from/to ***Producers, Integrators,*** *and* ***Consumers***

Multi-party acquisition and OA development ecosystems



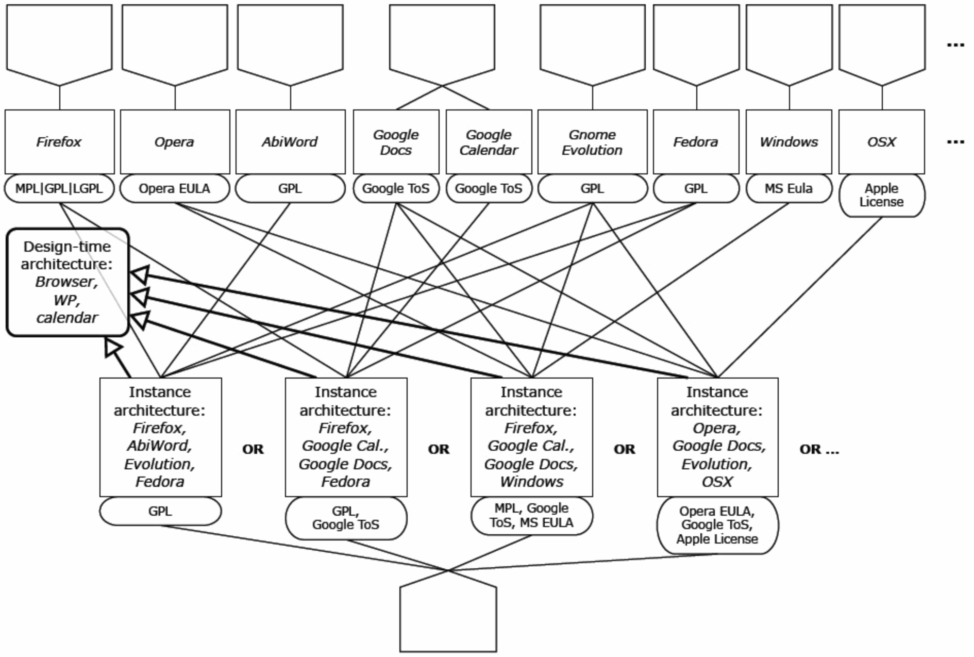
A sample elaboration of *producers* (vendors), software component *applications*, and *IP licenses* for OA system components.

Design-time view of an OA system for Mobile Devices



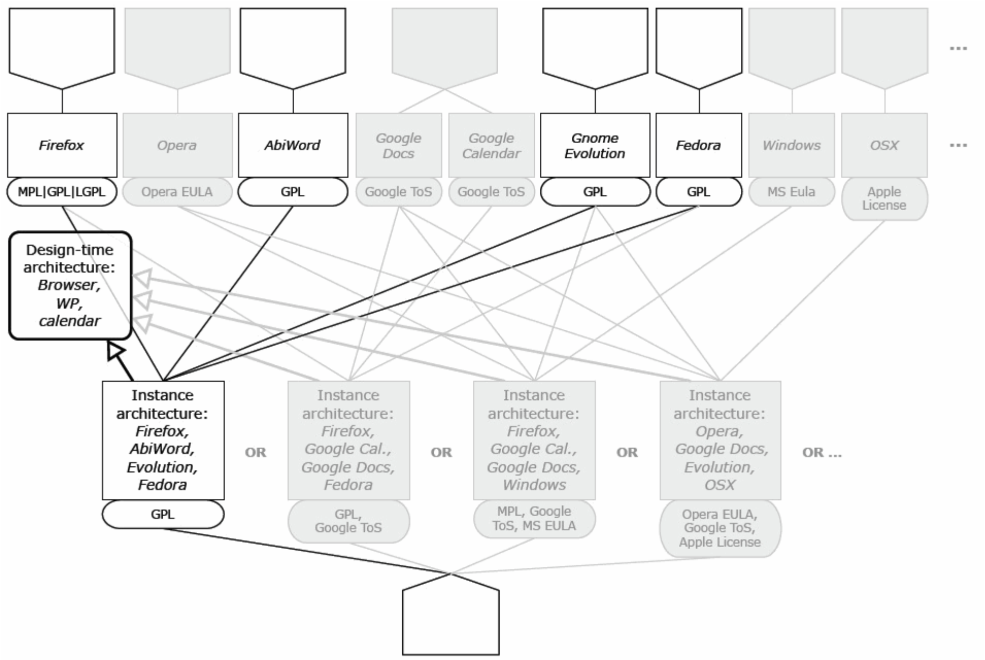
7

Software product line of *functionally similar* OA system alternatives


...
Firefox
Opera
AbiWord
Google Docs
Google Calendar
Gnome Evolution
Fedora
Windows
OSX
...
MPL|GPL|LGPL
Opera EULA
GPL
Google ToS
Google ToS
GPL
GPL
MS Eula
Apple License
Design-time architecture: Browser, WP, calendar
Instance architecture: Firefox, AbiWord, Evolution, Fedora
OR
Instance architecture: Firefox, Google Cal., Google Docs, Fedora
OR
Instance architecture: Firefox, Google Cal., Google Docs, Windows
OR
Instance architecture: Opera, Google Docs, Evolution, OSX
OR ...
GPL
GPL, Google ToS
MPL, Google ToS, MS EULA
Opera EULA, Google ToS, Apple License

8

Product line selection of one alternative system configuration



...

Firefox
Opera
AbiWord
Google Docs
Google Calendar
Gnome Evolution
Fedora
Windows
OSX
...

MPL|GPL|LGPL
Opera EULA
GPL
Google ToS
Google ToS
GPL
GPL
MS Eula
Apple License

Design-time architecture: *Browser, WP, calendar*

Instance architecture: *Firefox, AbiWord, Evolution, Fedora*
GPL

OR

Instance architecture: *Firefox, Google Cal., Google Docs, Fedora*
GPL, Google ToS

OR

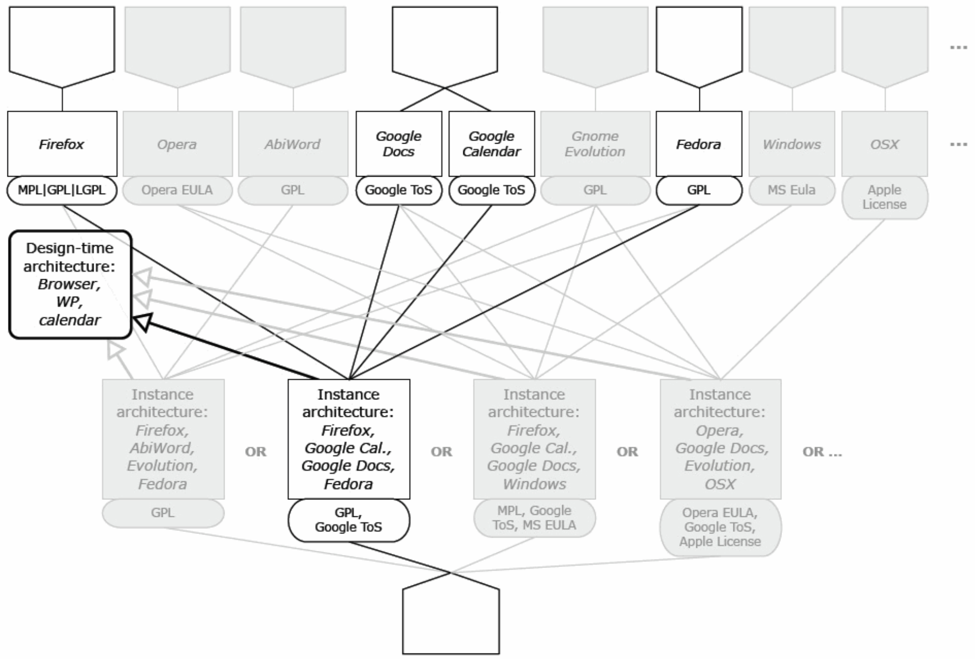
Instance architecture: *Firefox, Google Cal., Google Docs, Windows*
MPL, Google ToS, MS EULA

OR

Instance architecture: *Opera, Google Docs, Evolution, OSX*
Opera EULA, Google ToS, Apple License

OR ...

Product line selection of different functionally similar alternative



...

Firefox
Opera
AbiWord
Google Docs
Google Calendar
Gnome Evolution
Fedora
Windows
OSX
...

MPL|GPL|LGPL
Opera EULA
GPL
Google ToS
Google ToS
GPL
GPL
MS Eula
Apple License

Design-time architecture: *Browser, WP, calendar*

Instance architecture: *Firefox, AbiWord, Evolution, Fedora*
GPL

OR

Instance architecture: *Firefox, Google Cal., Google Docs, Fedora*
GPL, Google ToS

OR

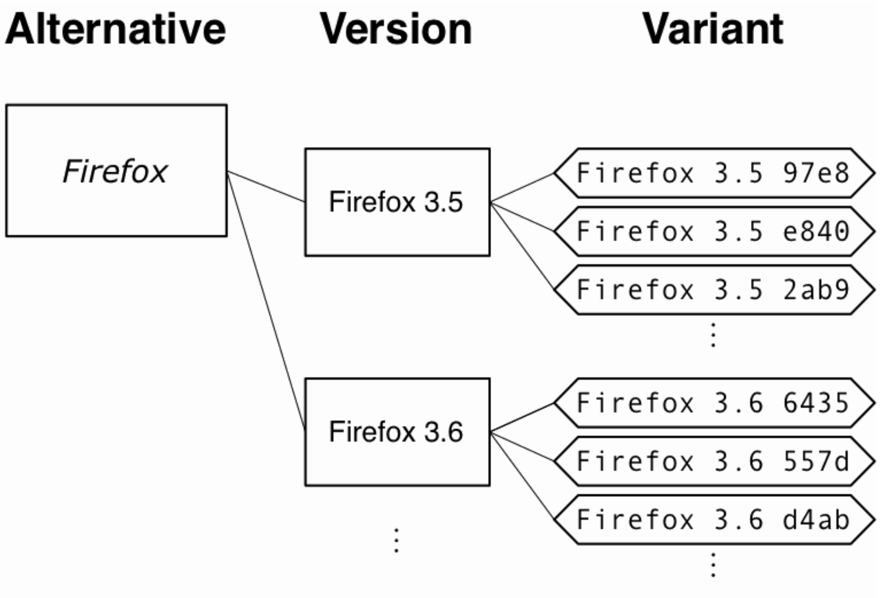
Instance architecture: *Firefox, Google Cal., Google Docs, Windows*
MPL, Google ToS, MS EULA

OR

Instance architecture: *Opera, Google Docs, Evolution, OSX*
Opera EULA, Google ToS, Apple License

OR ...

One Web browser component alternative, versions, and instance variants for inclusion via dynamic reconfiguration of OA system



Alternative | Version | Variant

Firefox

Firefox 3.5

Firefox 3.5 97e8

Firefox 3.5 e840

Firefox 3.5 2ab9

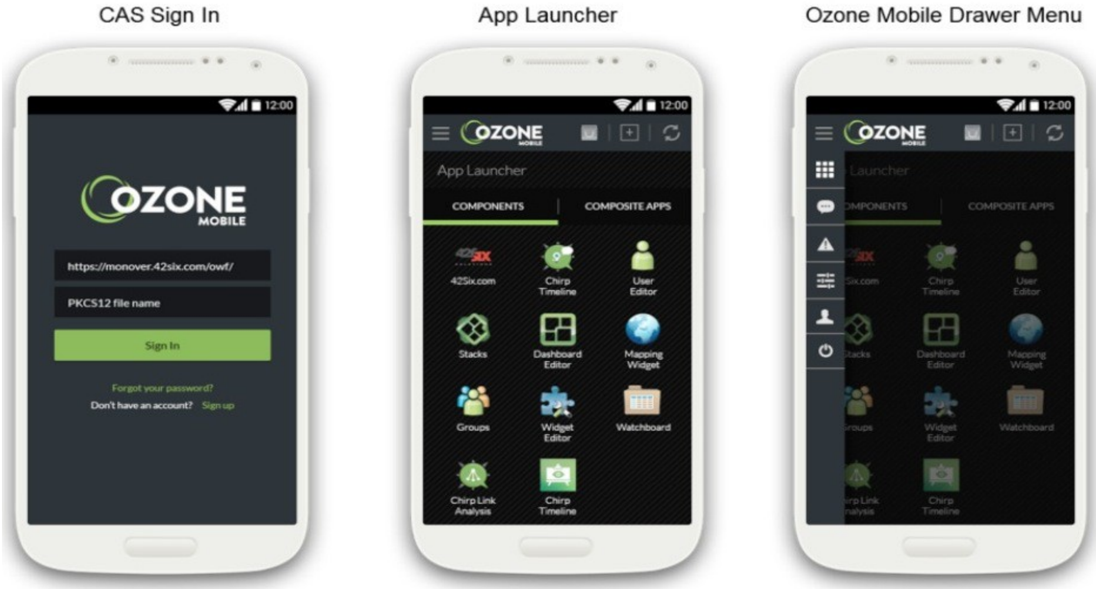
Firefox 3.6

Firefox 3.6 6435

Firefox 3.6 557d

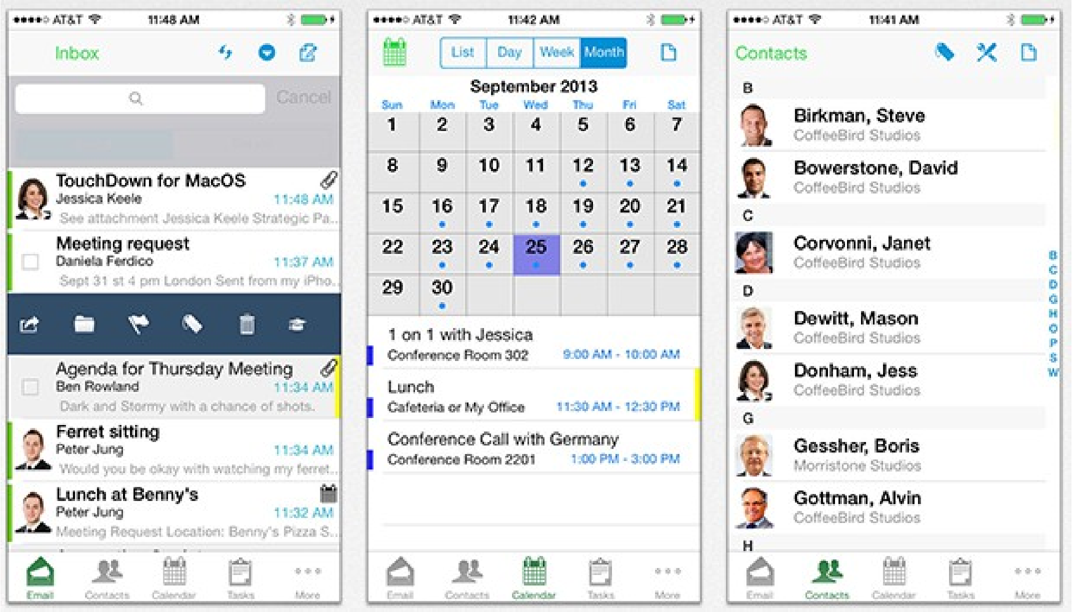
Firefox 3.6 d4ab

Shared development of Apps and Widgets as OA system components



Ozone Widget Framework for Web PCs and Mobile Devices

Mobile Apps/Widgets (plus middleware services, not shown)



TouchDown for MacOS
Jessica Keele 11:48 AM
See attachment Jessica Keele Strategic Pa...

Meeting request
Daniela Ferdico 11:37 AM
Sept 31 st 4 pm London Sent from my iPho...

Agenda for Thursday Meeting
Ben Rowland 11:34 AM
Dark and Stormy with a chance of shots.

Ferret sitting
Peter Jung 11:34 AM
Would you be okay with watching my ferret...

Lunch at Benny's
Peter Jung 11:32 AM
Meeting Request Location: Benny's Pizza S...

September 2013

Sun	Mon	Tue	Wed	Thu	Fri	Sat
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30					

1 on 1 with Jessica
Conference Room 302 9:00 AM - 10:00 AM

Lunch
Cafeteria or My Office 11:30 AM - 12:30 PM

Conference Call with Germany
Conference Room 2201 1:00 PM - 3:00 PM

Birkman, Steve
CoffeeBird Studios

Bowerstone, David
CoffeeBird Studios

Corvonni, Janet
CoffeeBird Studios

Dewitt, Mason
CoffeeBird Studios

Donham, Jess
CoffeeBird Studios

Gessher, Boris
Morristone Studios

Gottman, Alvin
CoffeeBird Studios

Mobile Middleware IP Licenses (for NitroDesk *Touchdown*)

LGPL 2.1

Ical4j from Ben Fortuna

Public Domain Declaration

Apache 2

The Legion of the Bouncy Castle

Creative Commons BY

Sony Mobile

Jesse Anderson

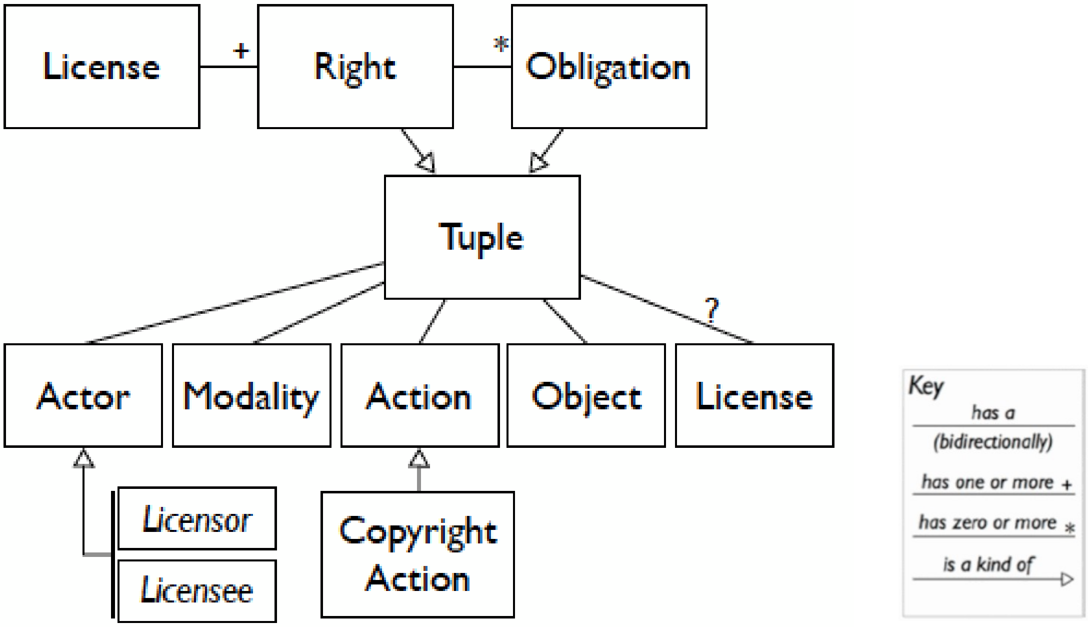
OpenSSL

Apple Non-Exclusive

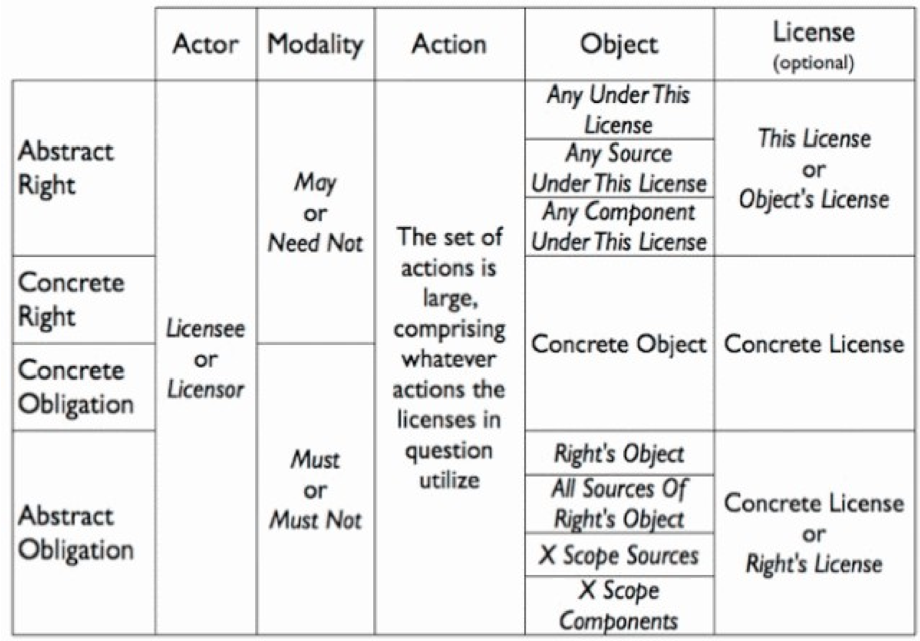
SQLite

Microsoft Public License

Software license meta-model for specifying constraint annotations

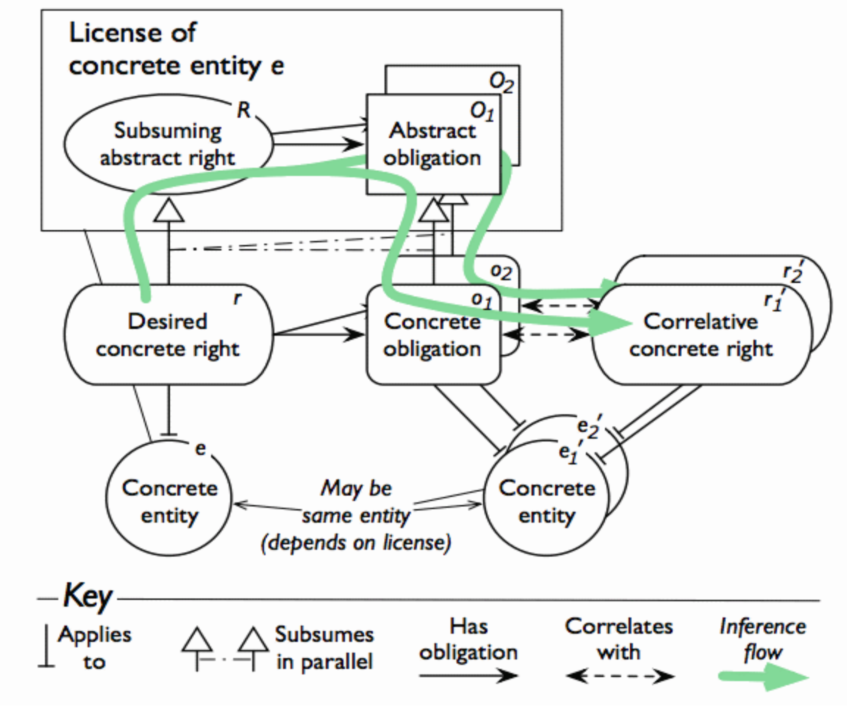


Logical modality and objects of software license rights and obligations constraints

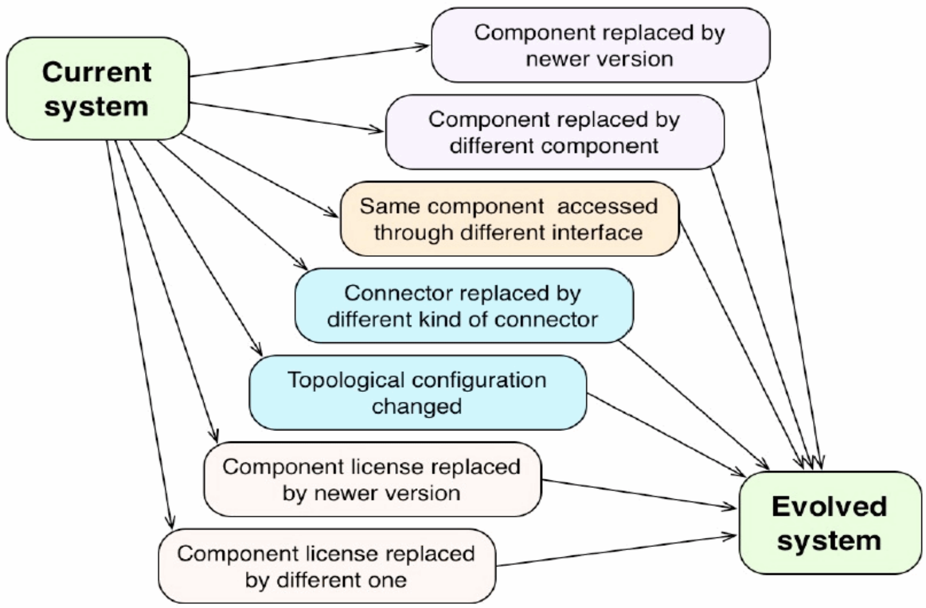


	Actor	Modality	Action	Object	License (optional)
Abstract Right	Licensee or Licensor	May or Need Not	The set of actions is large, comprising whatever actions the licenses in question utilize	Any Under This License Any Source Under This License Any Component Under This License	This License or Object's License
Concrete Right				Concrete Object	Concrete License
Concrete Obligation		Must or Must Not			
Abstract Obligation				Right's Object All Sources Of Right's Object X Scope Sources X Scope Components	Concrete License or Right's License

License inference scheme



Multi-party acquisition and OA development ecosystems: *Multiple OA system evolution paths*


Current system
Component replaced by newer version
Component replaced by different component
Same component accessed through different interface
Connector replaced by different kind of connector
Topological configuration changed
Component license replaced by newer version
Component license replaced by different one
Evolved system

IP and cybersecurity requirements will need continuous attention!

New business models for OA software components

- Franchising
- Enterprise licensing
- Metered usage
- Advertising supported
- Subscription
- Free component, paid service fees
- Federated reciprocity for shared development
- Collaborative buying
- Sponsorship/donation
- (Government) open source software
- and others

Managing acquisition costs will be demanding. Acquisition workforce will need automated assistance*, else acquisition management costs will dominate development costs for OA software components!*

Emerging challenges in achieving BBP via OA software systems

- Acquisition program managers/staff *may not understand* how software IP licenses affect OA system design, and vice-versa.
- Software IP and cybersecurity obligations and rights propagate across system development, deployment, and evolution activities *in ways not well understood* by system developers, integrators, end-users, or acquisition managers.

Emerging challenges in achieving BBP via OA software systems

- *Failure to understand* software IP and cybersecurity obligations and rights propagation can reduce DoD buying power, increase software life cycle costs, and reduce competition.
- DoD and other Government agencies *would financially and administratively benefit* from engaging the development and deployment of an (open source) automated software obligations and rights management system for the acquisition workforce.

New practices to realize cost-effective acquisition of OA software systems

- Need to R&D ***worked examples*** of reference OA system models, and component evolution alternatives.
- Need ***open source models of*** app/widget security assurance ***processes and*** reusable cybersecurity ***requirements.***
- Need precise ***domain-specific languages*** (DSLs) and ***automated analysis tools*** for continuously assessing and continuously improving cybersecurity and IP requirements for OA C2 systems composed from apps/widgets.

Conclusions

- Our research identifies how new software component technologies, IP and security requirements, and new business models interact to drive-down or drive-up acquisition costs.
- New technical risks for component-based OA software systems can dilute the cost-effectiveness of BBP efforts.
- Need R&D leading to automated systems that can model and analyze OA system IP licenses and cybersecurity requirements
  - Empower the acquisition workforce
  - Identify and manage cost-effectiveness trade-offs

Growing diversity of challenges in cybersecurity (another form of IP)

- Scacchi, W. and Alspaugh, T. (2012) Addressing Challenges in the Acquisition of Secure Software Systems with Open Architectures, *Proc. 9th Acquisition Research Symposium*, Vol. 1, 165-184, Naval Postgraduate School, Monterey, CA.
- Scacchi, W. and Alspaugh, T. (2013a). Processes in Securing Open Architecture Software Systems, *Proc. 2013 Intern. Conf. Software and System Processes*, San Francisco, CA, May 2013.
- Scacchi, W. and Alspaugh, T.A. (2013b). Streamlining the Process of Acquiring Secure Open Architecture Software Systems, *Proc. 10th Annual Acquisition Research Symposium*, Monterey, CA, 608-623, May 2013.
- Scacchi, W. and Alspaugh, T.A. (2013c). Challenges in the Development and Evolution of Secure Open Architecture Command and Control Systems, *Proc. 18th Intern. Command and Control Research and Technology Symposium*, Paper-098, Alexandria, VA, June 2013.

Acknowledgements

This material is based upon work supported by the Naval Postgraduate School Acquisition Research Program under Grant No. N00244-14-1-0030. The views expressed in materials or publications, and/or made by the presenters, do not necessarily reflect the official policies of the Naval Postgraduate School, nor does mentions of trade names, commercial practices, or organizations imply endorsement by the U.S. Government.

INSTITUTE *for* SOFTWARE RESEARCH
UNIVERSITY *of* CALIFORNIA • IRVINE

Chapter 4:

Reasoning about the Security of Open Architecture Software Systems for Mobile Devices

Reasoning about the Legal Issues in Open Architecture Software Systems for Mobile Devices

Walt Scacchi and Thomas Alspaugh
Institute for Software Research
University of California, Irvine
Irvine, CA 92697-3455 USA

August 2014

Overview

- Legal issues of interest
  - Intellectual Property (IP) licenses for software apps: *rights and obligations*
  - Vendor-Government *business models*
  - *Cybersecurity,* business models, IP *interactions*
- Reasoning about secure Open Architecture (OA) systems
- Examples of recent or work-in-progress results

Sample of our prior efforts in developing languages for reasoning about features of software system architectures

Choi, S. and Scacchi, W. (2003). Formal Analysis of the Structural Correctness of Software Life Cycle Descriptions, *Intern. J. Computers & Applications*, 25(2), 91-97.

Alspaugh, T.A., Scacchi, W., and Asuncion, H.A. (2010). Software Licenses in Context: The Challenge of Heterogeneously Licensed Systems, *J. Association for Information System*s, 11(11), 730-755.

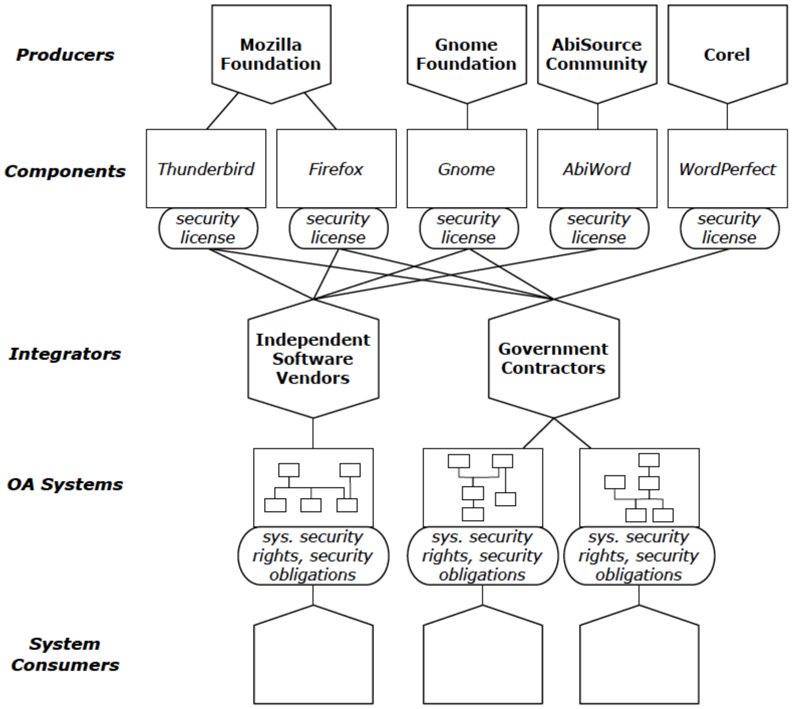
Scacchi, W. and Alspaugh, T.A. (2012). Understanding the Role of Licenses and Evolution in Open Architecture Software Ecosystems, *Journal of Systems and Software*, 85(7), 1479-1494.

Scacchi, W. and Alspaugh, T.A. (2013). Advances in the Acquisition of Secure Systems Based on Open Architectures, *Cyber Security and Information Systems J.* 1(2).

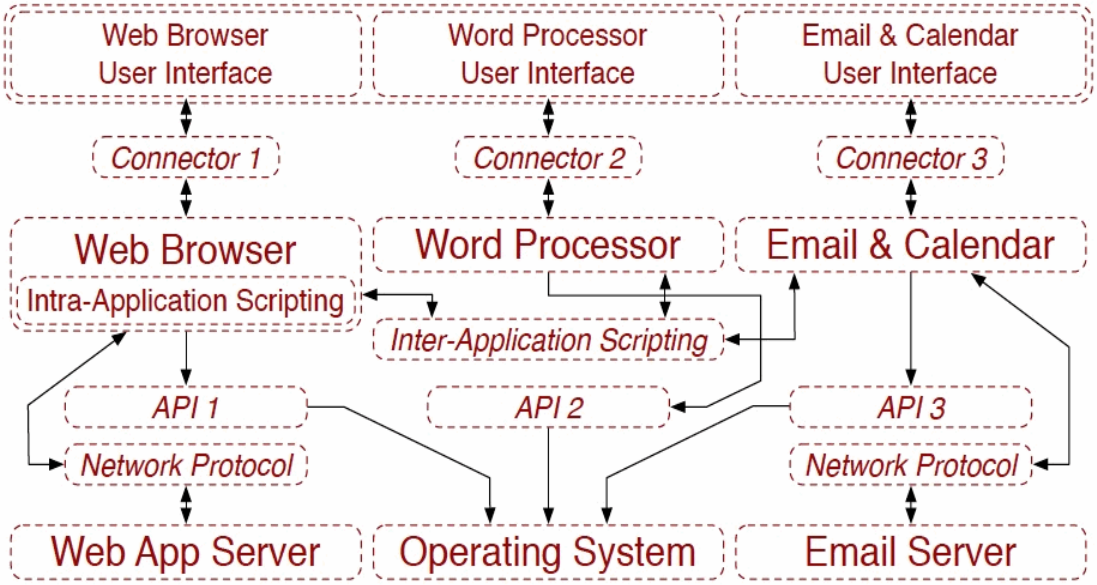
Examples of recent or work-in-progress results

- Language for modeling and meta-modeling as basis for reasoning about OA obligations and rights (simple constraints—not security)
- Tools and techniques for automated reasoning about OA obligations and rights integrity
- OA system software development environment
  - Based on *Eclipse* with *UCI ArchStudio* and analysis plug-in modules

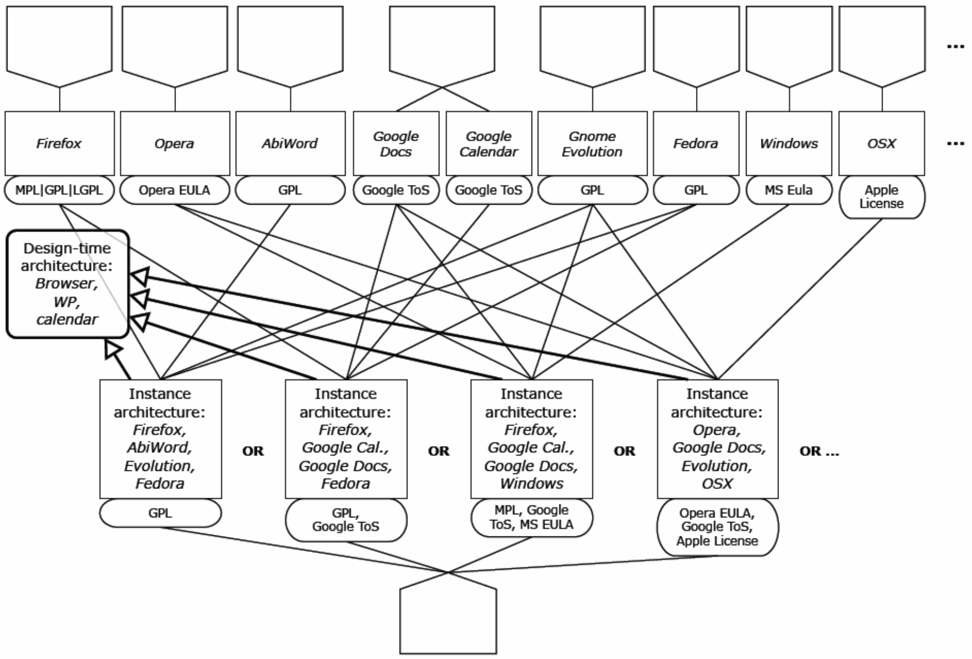
Software ecosystem of OA system producers, integrators, consumers



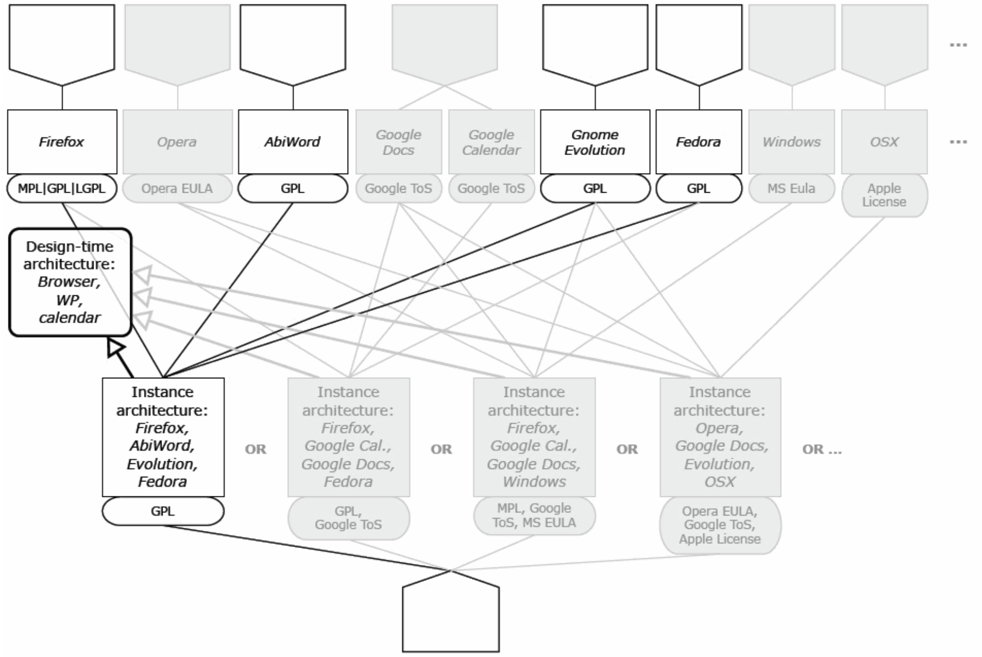
Design-time view of an OA system for Mobile Devices



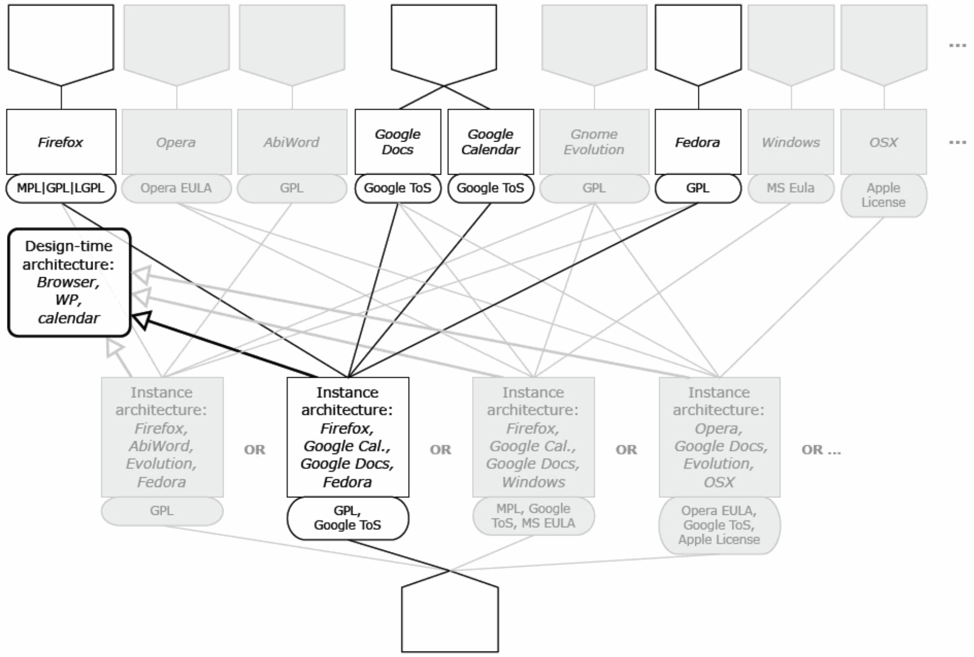
Software product line of *functionally similar* OA system alternatives


...
Firefox
Opera
AbiWord
Google Docs
Google Calendar
Gnome Evolution
Fedora
Windows
OSX
...
MPL|GPL|LGPL
Opera EULA
GPL
Google ToS
Google ToS
GPL
GPL
MS Eula
Apple License
Design-time architecture: Browser, WP, calendar
Instance architecture: Firefox, AbiWord, Evolution, Fedora
GPL
OR
Instance architecture: Firefox, Google Cal., Google Docs, Fedora
GPL, Google ToS
OR
Instance architecture: Firefox, Google Cal., Google Docs, Windows
MPL, Google ToS, MS EULA
OR
Instance architecture: Opera, Google Docs, Evolution, OSX
Opera EULA, Google ToS, Apple License
OR ...

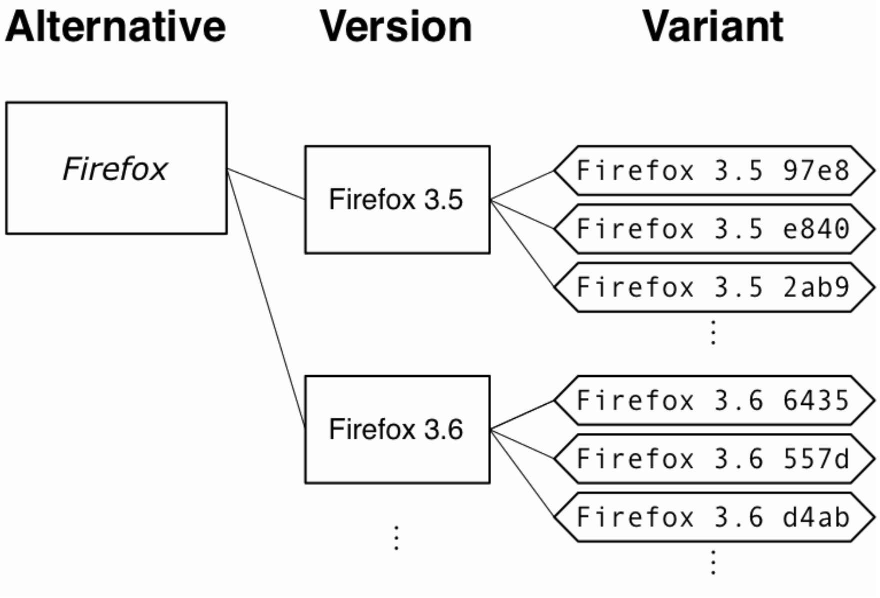
Product line selection of one alternative system configuration


...
Firefox
Opera
AbiWord
Google Docs
Google Calendar
Gnome Evolution
Fedora
Windows
OSX
...
MPL|GPL|LGPL
Opera EULA
GPL
Google ToS
Google ToS
GPL
GPL
MS Eula
Apple License
Design-time architecture: Browser, WP, calendar
Instance architecture: Firefox, AbiWord, Evolution, Fedora
GPL
OR
Instance architecture: Firefox, Google Cal., Google Docs, Fedora
GPL, Google ToS
OR
Instance architecture: Firefox, Google Cal., Google Docs, Windows
MPL, Google ToS, MS EULA
OR
Instance architecture: Opera, Google Docs, Evolution, OSX
Opera EULA, Google ToS, Apple License
OR ...

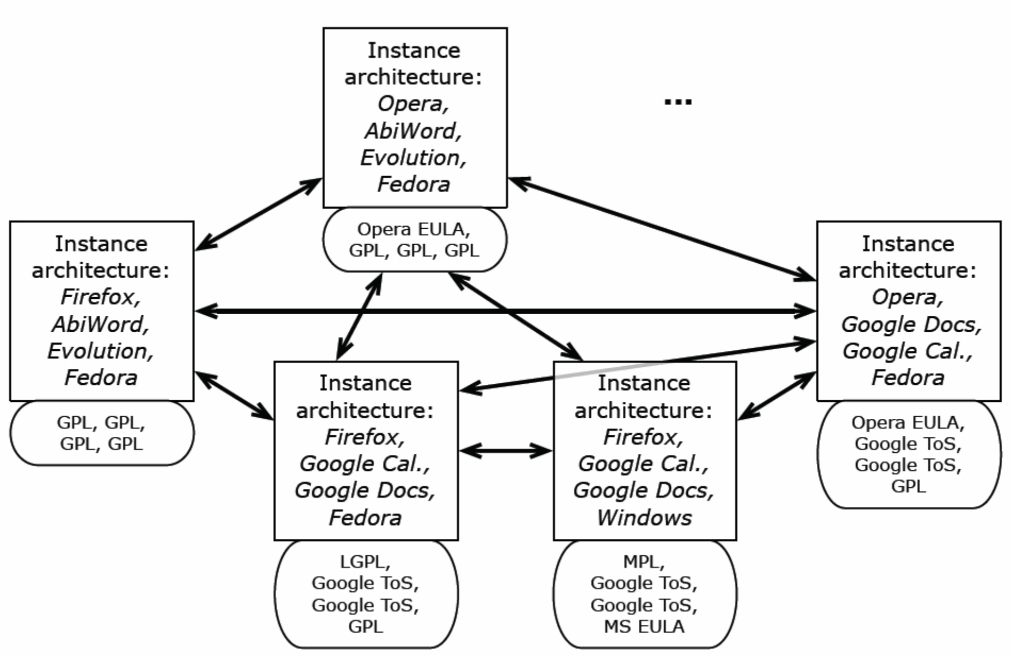
Product line selection of different functionally similar alternative


...
Firefox
Opera
AbiWord
Google Docs
Google Calendar
Gnome Evolution
Fedora
Windows
OSX
...
MPL|GPL|LGPL
Opera EULA
GPL
Google ToS
Google ToS
GPL
GPL
MS Eula
Apple License
Design-time architecture: Browser, WP, calendar
Instance architecture: Firefox, AbiWord, Evolution, Fedora
GPL
OR
Instance architecture: Firefox, Google Cal., Google Docs, Fedora
GPL, Google ToS
OR
Instance architecture: Firefox, Google Cal., Google Docs, Windows
MPL, Google ToS, MS EULA
OR
Instance architecture: Opera, Google Docs, Evolution, OSX
Opera EULA, Google ToS, Apple License
OR ...

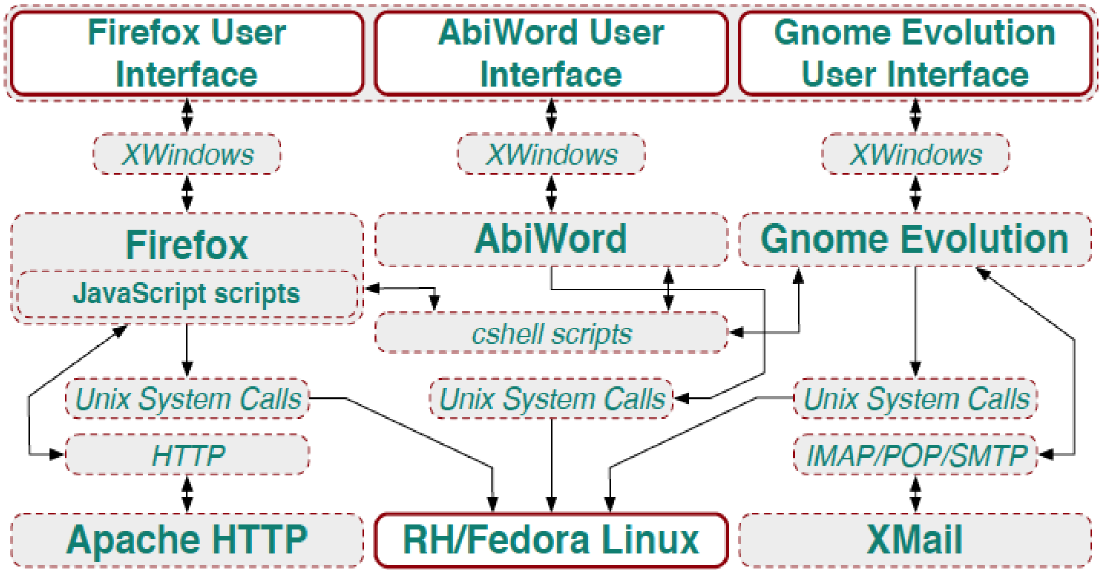
One Web browser component alternative, versions, and instance variants for inclusion via dynamic reconfiguration of OA system



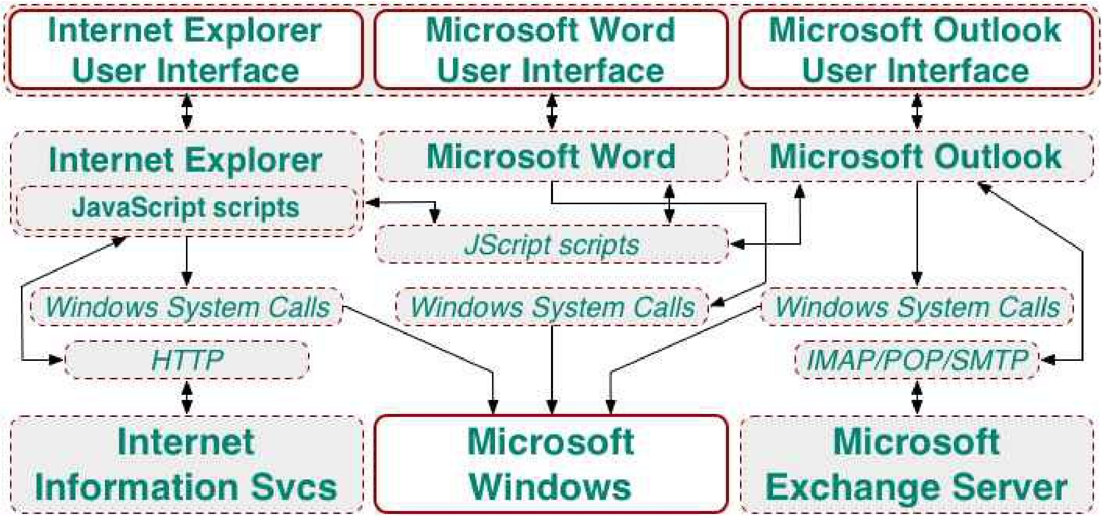
Partial view product family member instance variations for *dynamic reconfiguration* of OA system


Instance architecture: *Opera, AbiWord, Evolution, Fedora*
Opera EULA, GPL, GPL, GPL
...
Instance architecture: *Firefox, AbiWord, Evolution, Fedora*
GPL, GPL, GPL, GPL
Instance architecture: *Opera, Google Docs, Google Cal., Fedora*
Opera EULA, Google ToS, Google ToS, GPL
Instance architecture: *Firefox, Google Cal., Google Docs, Fedora*
LGPL, Google ToS, Google ToS, GPL
Instance architecture: *Firefox, Google Cal., Google Docs, Windows*
MPL, Google ToS, Google ToS, MS EULA

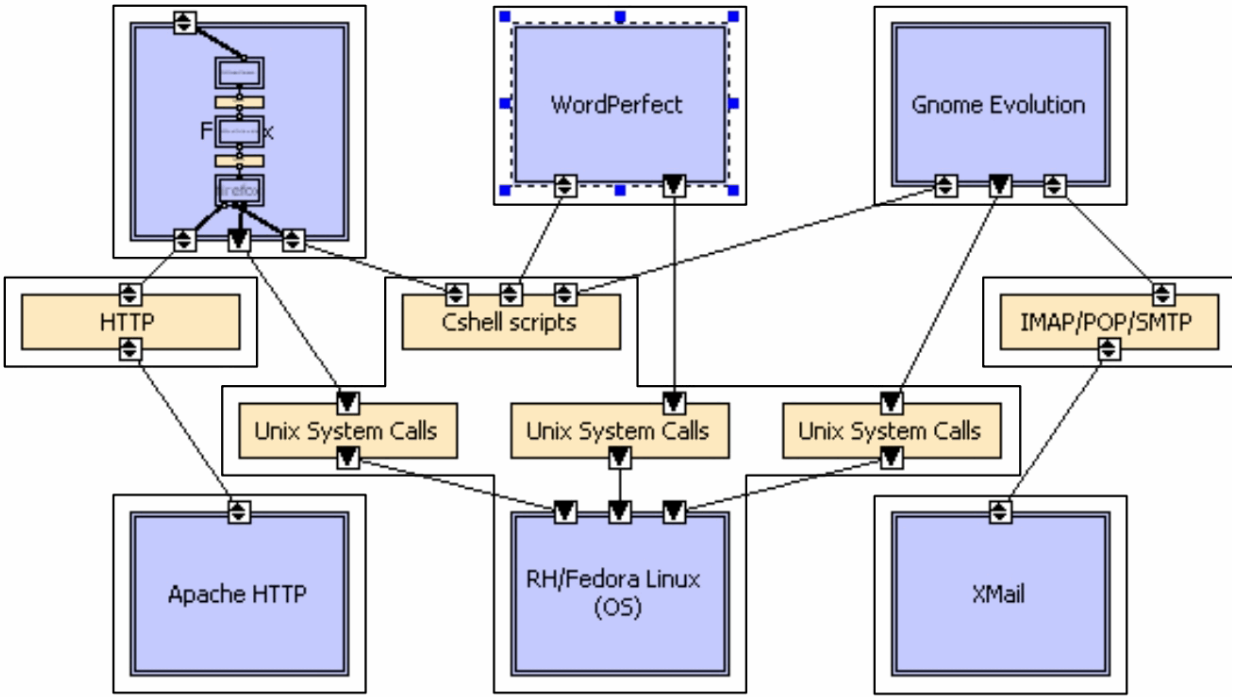
Build-time view of OA design selecting *OSS* product family alternatives



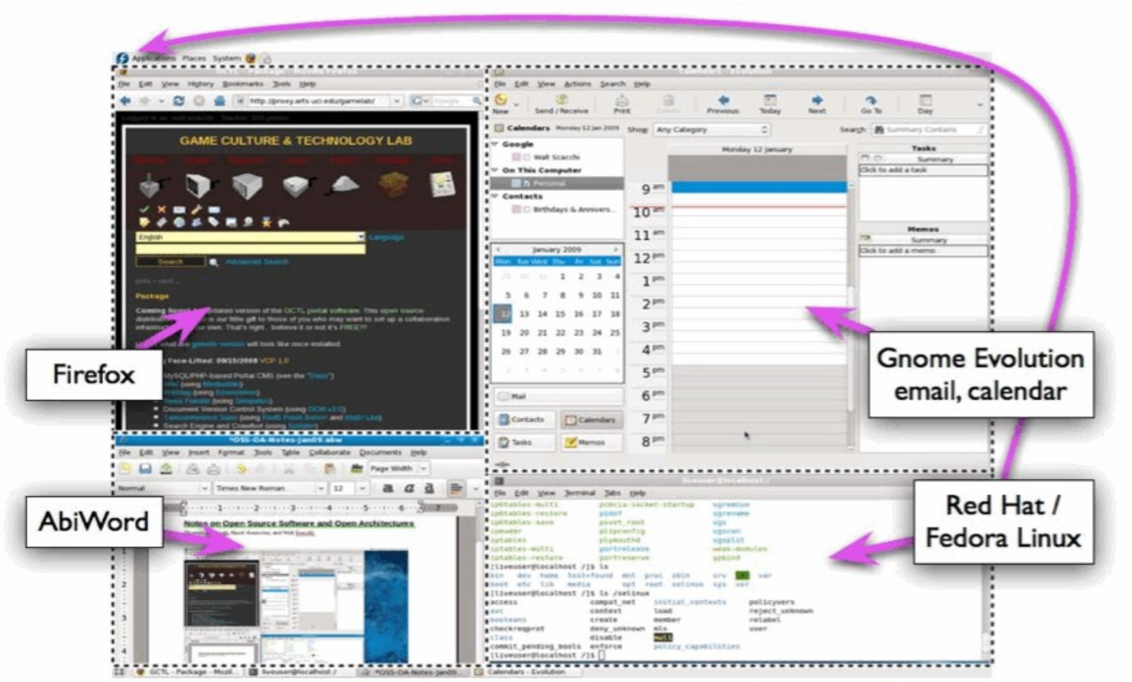
Build-time view of OA design selecting *proprietary* product family alternatives



Build-time view of an OA system security encapsulation scheme

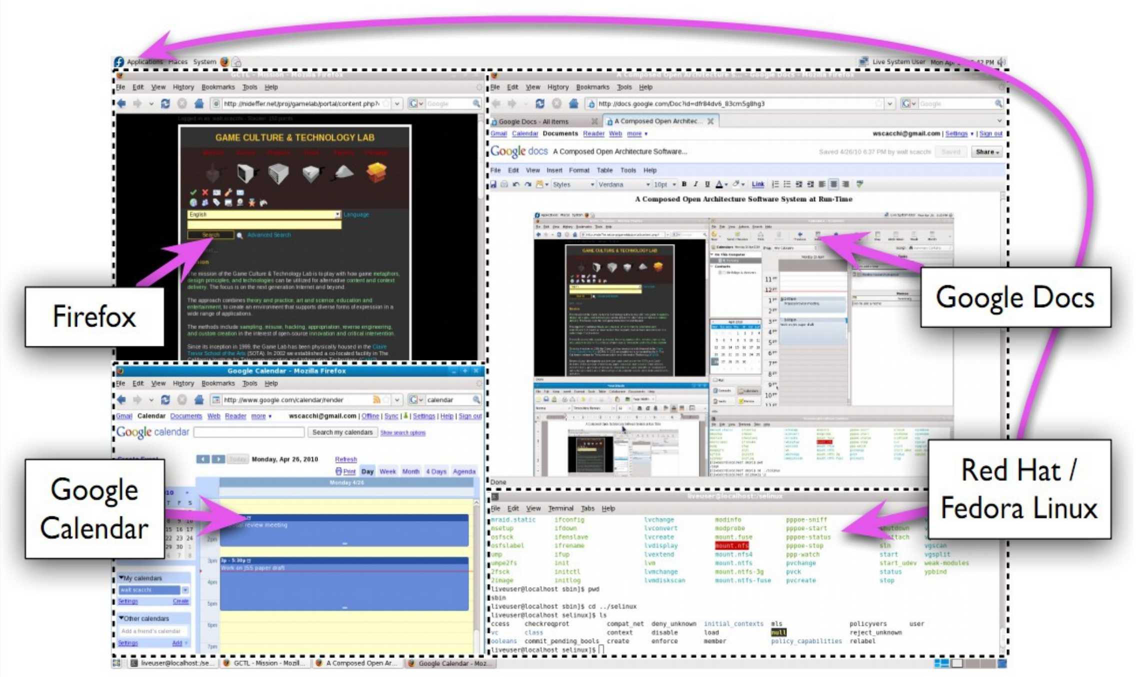


Run-time deployment view of OA system family member configuration


Firefox
AbiWord
Gnome Evolution email, calendar
Red Hat / Fedora Linux
GAME CULTURE & TECHNOLOGY LAB

```
ip6tables-multi        pcmcia-socket-startup    vgremove
ip6tables-restore      pidof                    vgrename
ip6tables-save         pivot_root               vgs
ipmaddr                plipconfig               vgscan
iptables               plymouthd                vgsplit
iptables-multi         portrelease              weak-modules
iptables-restore       portreserve              ypbind
[liveuser@localhost /]$ ls
bin   dev  home  lost+found  mnt  proc  sbin     srv  tmp  var
boot  etc  lib   media       opt  root  selinux  sys  usr
[liveuser@localhost /]$ ls /selinux
access                compat_net    initial_contexts     policyvers
avc                   context       load                 reject_unknown
booleans              create        member               relabel
checkreqprot          deny_unknown  mls                  user
class                 disable       null
commit_pending_bools  enforce       policy_capabilities
[liveuser@localhost /]$
```

Run-time deployment view of a similar alternative OA system configuration

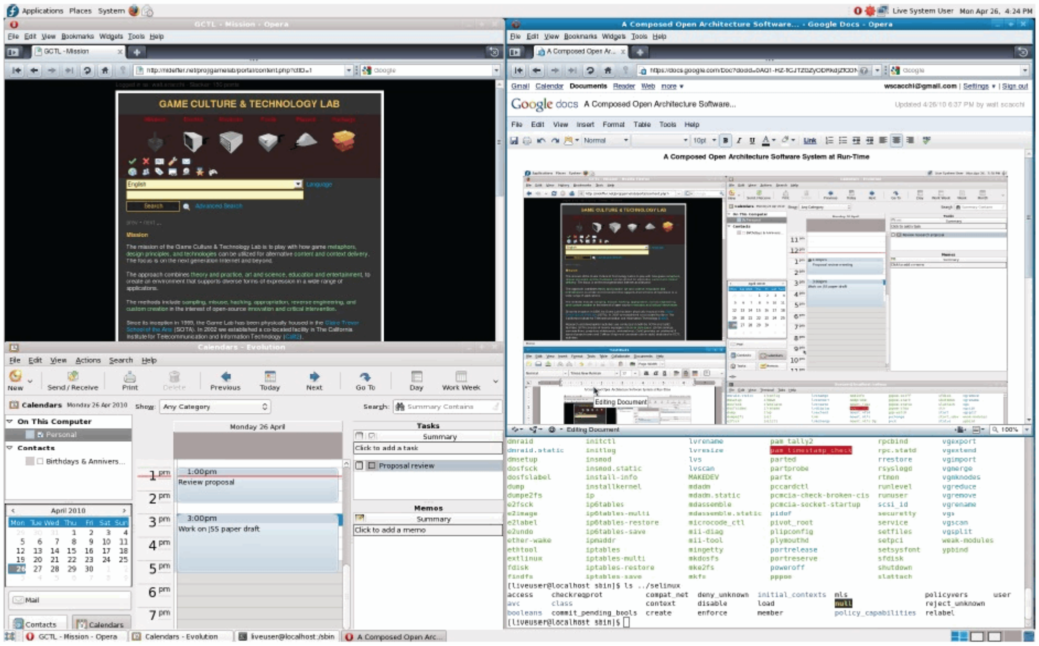

Firefox
Google Calendar
Google Docs
Red Hat / Fedora Linux
GAME CULTURE & TECHNOLOGY LAB
A Composed Open Architecture Software System at Run-Time

```
liveuser@localhost:/selinux
mraid.static   ifconfig     lvchange      modinfo          pppoe-sniff
msetup         ifdown       lvconvert     modprobe         pppoe-start        shutdown
osfsck         ifenslave    lvcreate      mount.fuse       pppoe-status       ...ttach
osfslabel      ifrename     lvdisplay     mount.nfs        pppoe-stop         sln      vgscan
ump            ifup         lvextend      mount.nfs4       ppp-watch          start    vgsplit
umpe2fs        init         lvm           mount.ntfs       pvchange           start_udev  weak-modules
2fsck          initctl      lvmchange     mount.ntfs-3g    pvck               status   ypbind
2image         initlog      lvmdiskscan   mount.ntfs-fuse  pvcreate           stop
liveuser@localhost sbin]$ pwd
sbin
liveuser@localhost sbin]$ cd ../selinux
liveuser@localhost selinux]$ ls
ccess    checkreqprot          compat_net  deny_unknown  initial_contexts  mls                  policyvers      user
vc       class                 context     disable       load              null                 reject_unknown
ooleans  commit_pending_bools  create      enforce       member            policy_capabilities  relabel
liveuser@localhost selinux]$
```

Types of evolutionary changes in OA systems that also change system configurations

- Component (version) evolution
- Component replacement by similar alternative
- Architecture evolution
  - Including dynamic reconfiguration
- Component security policy license evolution
  - Licenses represent *annotated constraints* on OA components, connectors, and configurations
- Change in desired license rights or acceptable obligations within an OA system

Run-time deployment view with alternative OA configuration

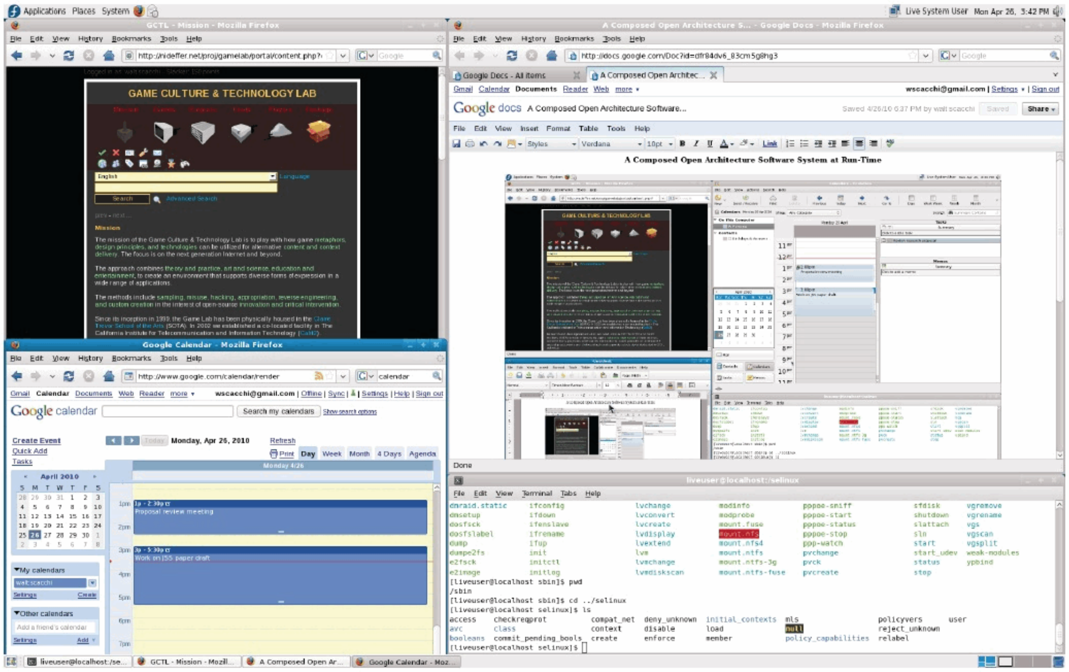

Applications Places System
GCTL - Mission - Opera
GAME CULTURE & TECHNOLOGY LAB
Calendars - Evolution
Calendars Monday 26 Apr 2010
On This Computer
Personal
Contacts
Birthdays & Anniversaries
April 2010
Monday 26 April
Tasks
Click to add a task
Proposal review
Memos
Click to add a memo
Live System User Mon Apr 26, 4:24 PM
A Composed Open Architecture Software... - Google Docs - Opera
Google docs A Composed Open Architecture Software...
wscacchi@gmail.com | Settings ▾ | Sign out
A Composed Open Architecture Software System at Run-Time
Editing Document
100%
GCTL - Mission - Opera
Calendars - Evolution
liveuser@localhost:/sbin
A Composed Open Arc...

1:00pm Review proposal

3:00pm Work on ISS paper draft

```
dmraid          initctl                lvrename             pam_tally2                rpcbind      vgexport
dmraid.static   initlog                lvresize             pam_timestamp_check       rpc.statd    vgextend
dmsetup         insmod                 lvs                  parted                    rrestore     vgimport
dosfsck         insmod.static          lvscan               partprobe                 rsyslogd     vgmerge
dosfslabel      install-info           MAKEDEV              partx                     rtmon        vgmknodes
dump            installkernel          mdadm                pccardctl                 runlevel     vgreduce
dumpe2fs        ip                     mdadm.static         pcmcia-check-broken-cis   runuser      vgremove
e2fsck          ip6tables              mdassemble           pcmcia-socket-startup     scsi_id      vgrename
e2image         ip6tables-multi        mdassemble.static    pidof                     securetty    vgs
e2label         ip6tables-restore      microcode_ctl        pivot_root                service      vgscan
e2undo          ip6tables-save         mii-diag             plipconfig                setfiles     vgsplit
ether-wake      ipmaddr                mii-tool             plymouthd                 setpci       weak-modules
ethtool         iptables               mingetty             portrelease               setsysfont   ypbind
extlinux        iptables-multi         mkdosfs              portreserve               sfdisk
fdisk           iptables-restore       mke2fs               poweroff                  shutdown
findfs          iptables-save          mkfs                 pppoe                     slattach
[liveuser@localhost sbin]$ ls ../selinux
access    checkreqprot          compat_net  deny_unknown  initial_contexts  mls                    policyvers     user
avc       class                 context     disable       load              null                   reject_unknown
booleans  commit_pending_bools  create      enforce       member            policy_capabilities    relabel
[liveuser@localhost sbin]$ 
```

Run-time deployment view with service-based OA configuration


Applications Places System
GCTL - Mission - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://nideffer.net/proj/gamelab/portal/content.php?
GAME CULTURE & TECHNOLOGY LAB
English
Language
Search
Advanced Search
prev - next ...
Google Calendar - Mozilla Firefox
http://www.google.com/calendar/render
Gmail Calendar Documents Web Reader more
wscacchi@gmail.com | Offline | Sync | Settings | Help | Sign out
Google calendar
Search my calendars
Show search options
Create Event
Quick Add
Tasks
Today
Monday, Apr 26, 2010
Refresh
Print Day Week Month 4 Days Agenda
April 2010
My calendars
Other calendars
A Composed Open Architecture S... - Google Docs - Mozilla Firefox
http://docs.google.com/Doc?id=dfr84dv6_83cm5g8hg3
Google Docs - All Items
A Composed Open Architec...
Google docs
A Composed Open Architecture Software...
Saved 4/26/10 3:37 PM by walt scacchi
Share
File Edit View Insert Format Table Tools Help
Styles
Verdana
10pt
Done
Live System User Mon Apr 26, 3:42 PM
liveuser@localhost:/selinux
File Edit View Terminal Tabs Help
liveuser@localhost:/se...
GCTL - Mission - Mozill...
A Composed Open Ar...
Google Calendar - Moz...

Mission

The mission of the Game Culture & Technology Lab is to play with how game metaphors, design principles, and technologies can be utilized for alternative content and context delivery. The focus is on the next generation Internet and beyond.

The approach combines theory and practice, art and science, education and entertainment, to create an environment that supports diverse forms of expression in a wide range of applications.

The methods include sampling, misuse, hacking, appropriation, reverse engineering, and custom creation in the interest of open-source innovation and critical intervention.

Since its inception in 1999, the Game Lab has been physically housed in the Claire Trevor School of the Arts (SOTA). In 2002 we established a co-located facility in The California Institute for Telecommunication and Information Technology (Calit2).

1p – 2:30p
Proposal review meeting

3p – 5:30p
Work on ISS paper draft

A Composed Open Architecture Software System at Run-Time

```
dmraid.static    ifconfig      lvchange      modinfo          pppoe-sniff   sfdisk       vgremove
dmsetup          ifdown        lvconvert     modprobe         pppoe-start   shutdown     vgrename
dosfsck          ifenslave     lvcreate      mount.fuse       pppoe-status  slattach     vgs
dosfslabel       ifrename      lvdisplay     mount.nfs        pppoe-stop    sln          vgscan
dump             ifup          lvextend      mount.nfs4       ppp-watch     start        vgsplit
dumpe2fs         init          lvm           mount.ntfs       pvchange      start_udev   weak-modules
e2fsck           initctl       lvmchange     mount.ntfs-3g    pvck          status       ypbind
e2image          initlog       lvmdiskscan   mount.ntfs-fuse  pvcreate      stop
[liveuser@localhost sbin]$ pwd
/sbin
[liveuser@localhost sbin]$ cd ../selinux
[liveuser@localhost selinux]$ ls
access    checkreqprot          compat_net  deny_unknown  initial_contexts  mls                   policyvers      user
avc       class                 context     disable       load              null                  reject_unknown
booleans  commit_pending_bools  create      enforce       member            policy_capabilities   relabel
[liveuser@localhost selinux]$ 
```

Combinatorially large space of alternative OA system configurations

- Example: OA system with 6 component types
  - Each component type defines a family of functionally similar alternatives
  - Assume three overall system platform alternatives
  - Each component type with 5 alternative producers
  - Each alternative providing 5 available versions
  - Each version providing 5 functionally equivalent variants
  - This allows for 6*3*5*5*5 = 2250 similar but distinct OA system configurations available for at build-time.

Combinatorially large space of alternative OA system configurations

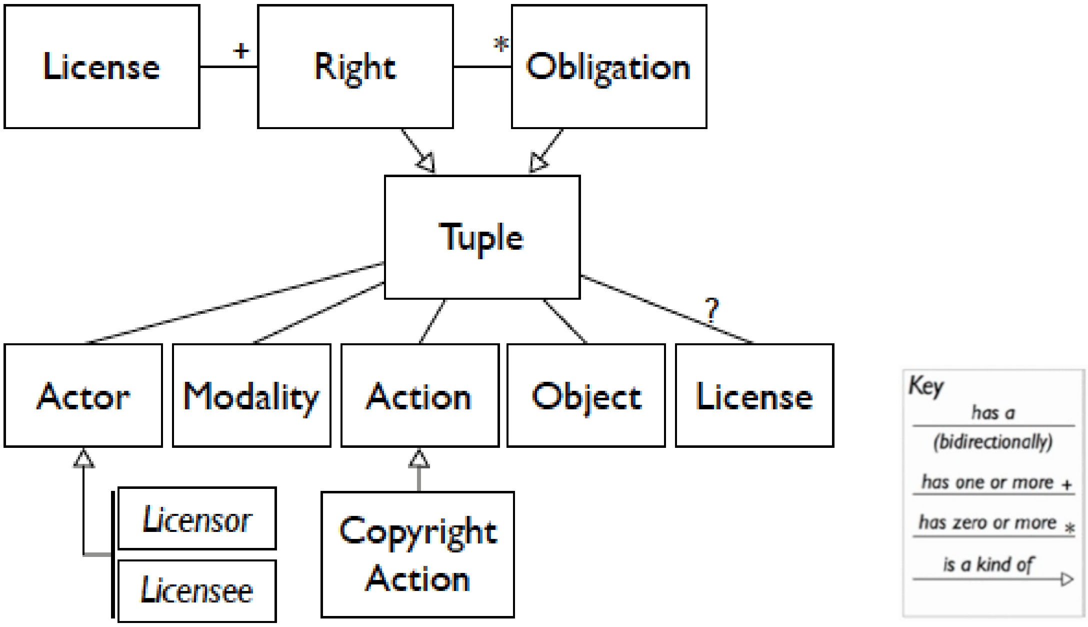
- Each configuration represents a specific attack surface, so *cost of attack grows combinatorially* with number of system alternatives to attack (i.e., which is the target now?)
- Cost of system defense via dynamic reconfiguration is low/constant
- Switching to alternative configurations can be handled via automated processes, and driven by *policy*, e.g.:
  - switch run-time configuration every 30 minutes;
  - provide concurrent users with similar system configurations
  - monitor and continuously cross-check different user configurations for problem (attack or corrupted) operation
    - If configuration is problematic, then randomly switch to alternative similar configuration
    - If configuration is OK, then switch to equivalent alternative configuration at start of new usage sessions.

Software security licenses, architectures, and analysis

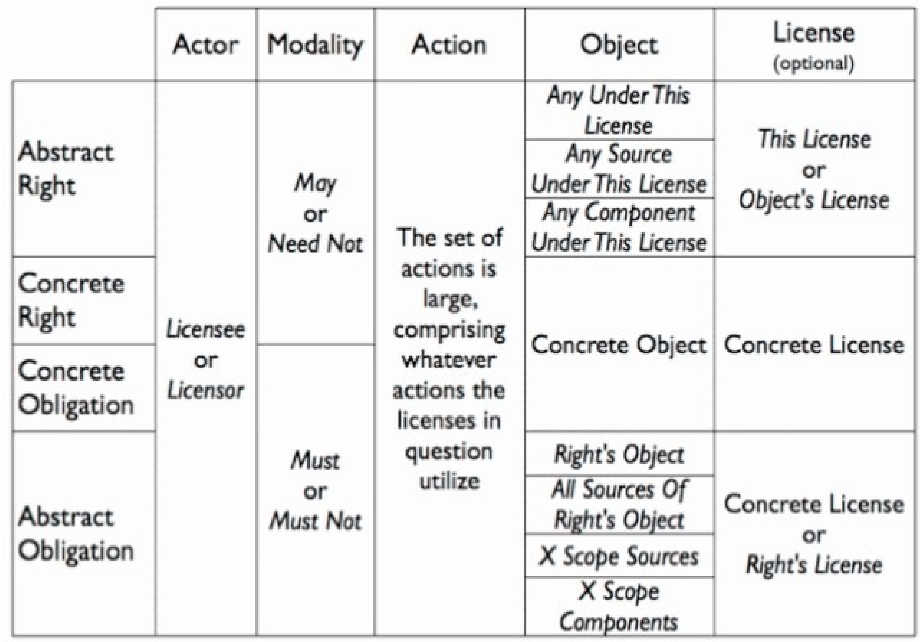
Specifying and analyzing system security requirements as "licenses"

- Security policies imply capabilities that correspond to *rights* and *obligations* in licenses
- Should be possible to specify and analyze system *security architecture* that conform to a *security meta-model*, much like we do for software IP licenses
- Should be possible to develop computational tools and development environments that can analyze security at *design-time*, *build-time*, and *run-time*, as well as at system *evolution-time*.

Software license meta-model for specifying constraint annotations

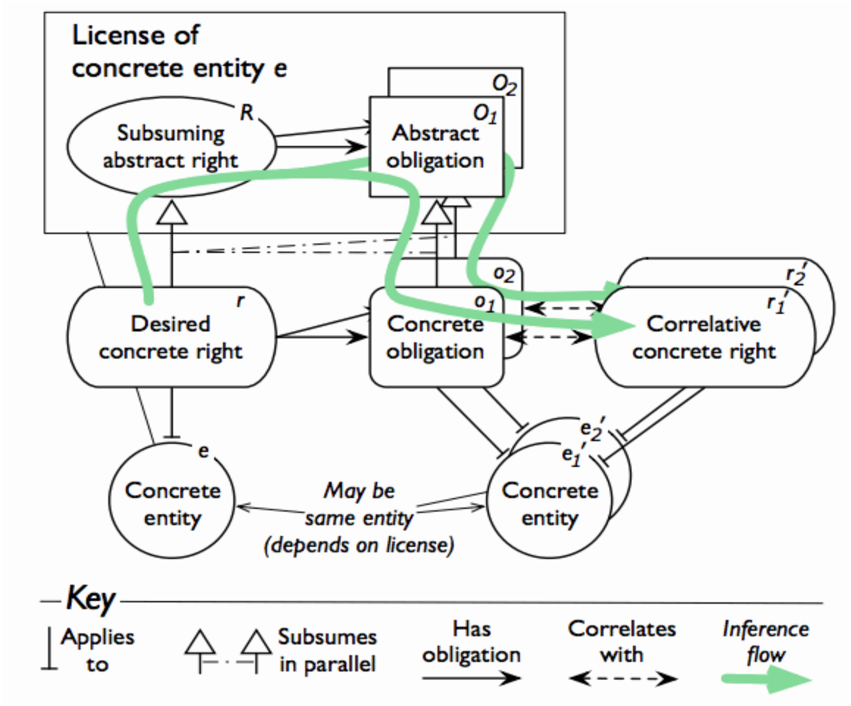


Logical modality and objects of software license rights and obligations constraints



	Actor	Modality	Action	Object	License (optional)
Abstract Right	*Licensee* or *Licensor*	*May* or *Need Not*	The set of actions is large, comprising whatever actions the licenses in question utilize	*Any Under This License* *Any Source Under This License* *Any Component Under This License*	*This License* or *Object's License*
Concrete Right				Concrete Object	Concrete License
Concrete Obligation		*Must* or *Must Not*			
Abstract Obligation				*Right's Object* *All Sources Of Right's Object* *X Scope Sources* *X Scope Components*	Concrete License or *Right's License*

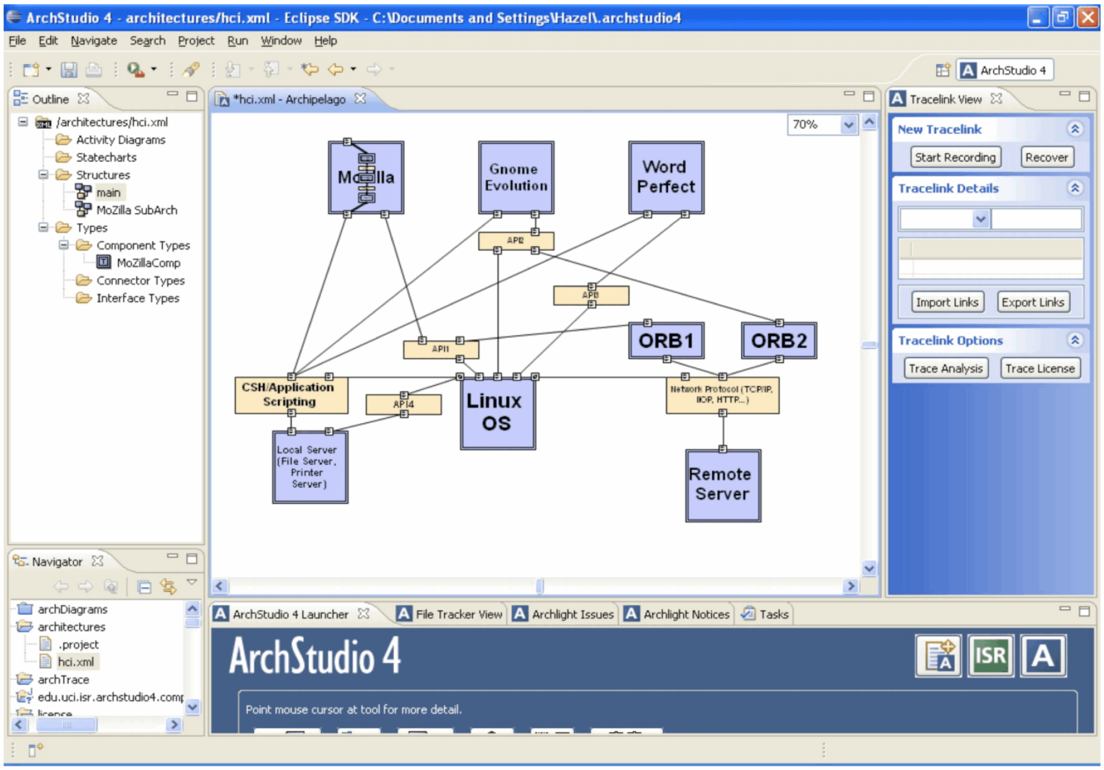
License inference scheme



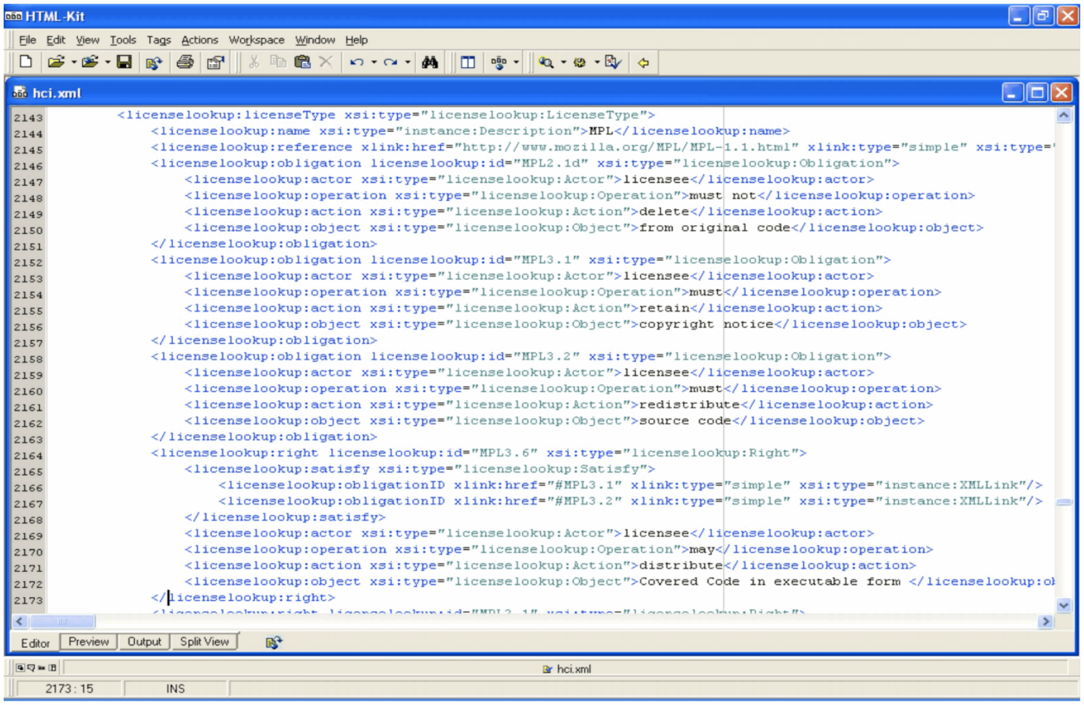
Security license analysis

- License types:
  - Strongly reciprocal; weakly reciprocal; academic; Terms of Service; Proprietary
  - “public domain” is not a type of license
- Propagation of reciprocal obligations
- Conflicting obligations
- Calculating obligations and rights

Prototype view of OA system development environment with license analysis plug-in

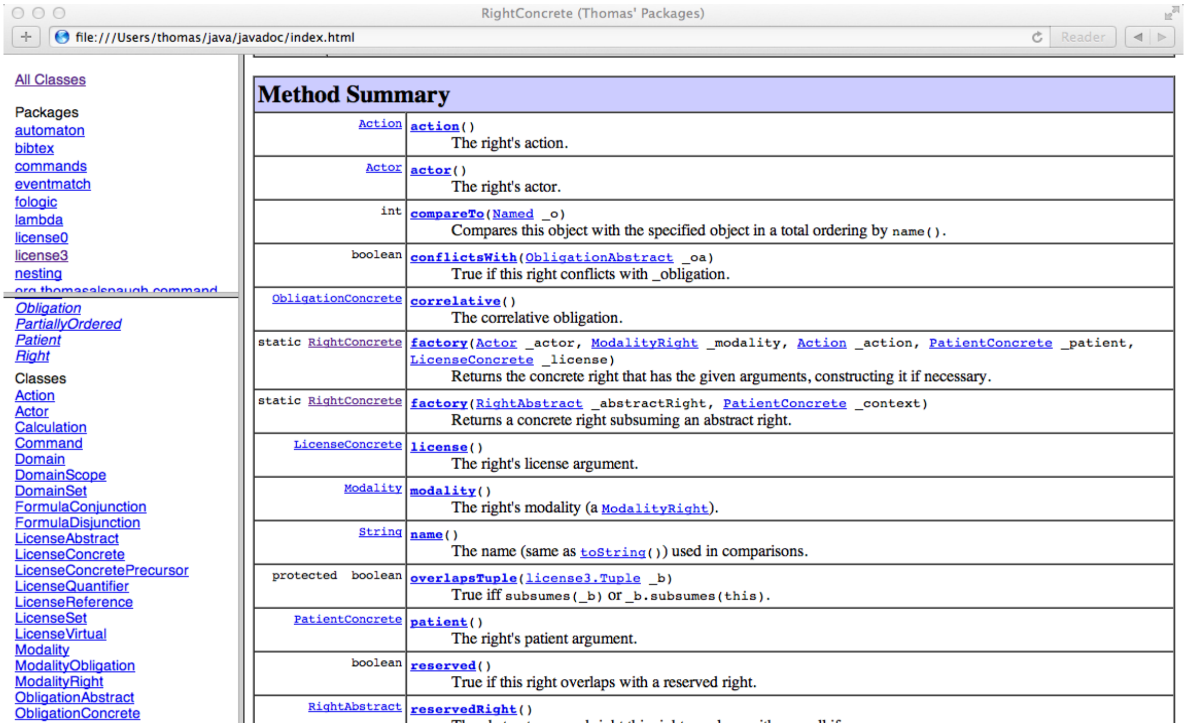


Internal form of component license annotation of OA prior to analysis


HTML-Kit
File Edit View Tools Tags Actions Workspace Window Help
hci.xml
Editor Preview Output Split View
hci.xml
2173 : 15
INS

```
<licenselookup:licenseType xsi:type="licenselookup:LicenseType">
    <licenselookup:name xsi:type="instance:Description">MPL</licenselookup:name>
    <licenselookup:reference xlink:href="http://www.mozilla.org/MPL/MPL-1.1.html" xlink:type="simple" xsi:type='
    <licenselookup:obligation licenselookup:id="MPL2.1d" xsi:type="licenselookup:Obligation">
        <licenselookup:actor xsi:type="licenselookup:Actor">licensee</licenselookup:actor>
        <licenselookup:operation xsi:type="licenselookup:Operation">must not</licenselookup:operation>
        <licenselookup:action xsi:type="licenselookup:Action">delete</licenselookup:action>
        <licenselookup:object xsi:type="licenselookup:Object">from original code</licenselookup:object>
    </licenselookup:obligation>
    <licenselookup:obligation licenselookup:id="MPL3.1" xsi:type="licenselookup:Obligation">
        <licenselookup:actor xsi:type="licenselookup:Actor">licensee</licenselookup:actor>
        <licenselookup:operation xsi:type="licenselookup:Operation">must</licenselookup:operation>
        <licenselookup:action xsi:type="licenselookup:Action">retain</licenselookup:action>
        <licenselookup:object xsi:type="licenselookup:Object">copyright notice</licenselookup:object>
    </licenselookup:obligation>
    <licenselookup:obligation licenselookup:id="MPL3.2" xsi:type="licenselookup:Obligation">
        <licenselookup:actor xsi:type="licenselookup:Actor">licensee</licenselookup:actor>
        <licenselookup:operation xsi:type="licenselookup:Operation">must</licenselookup:operation>
        <licenselookup:action xsi:type="licenselookup:Action">redistribute</licenselookup:action>
        <licenselookup:object xsi:type="licenselookup:Object">source code</licenselookup:object>
    </licenselookup:obligation>
    <licenselookup:right licenselookup:id="MPL3.6" xsi:type="licenselookup:Right">
        <licenselookup:satisfy xsi:type="licenselookup:Satisfy">
            <licenselookup:obligationID xlink:href="#MPL3.1" xlink:type="simple" xsi:type="instance:XMLLink"/>
            <licenselookup:obligationID xlink:href="#MPL3.2" xlink:type="simple" xsi:type="instance:XMLLink"/>
        </licenselookup:satisfy>
        <licenselookup:actor xsi:type="licenselookup:Actor">licensee</licenselookup:actor>
        <licenselookup:operation xsi:type="licenselookup:Operation">may</licenselookup:operation>
        <licenselookup:action xsi:type="licenselookup:Action">distribute</licenselookup:action>
        <licenselookup:object xsi:type="licenselookup:Object">Covered Code in executable form </licenselookup:ol
    </licenselookup:right>
```

Directory of computational methods for analyzing "rights"


RightConcrete (Thomas' Packages)
file:///Users/thomas/java/javadoc/index.html
Reader

All Classes

Packages
automaton
bibtex
commands
eventmatch
fologic
lambda
license0
license3
nesting
org.thomasalspaugh.command

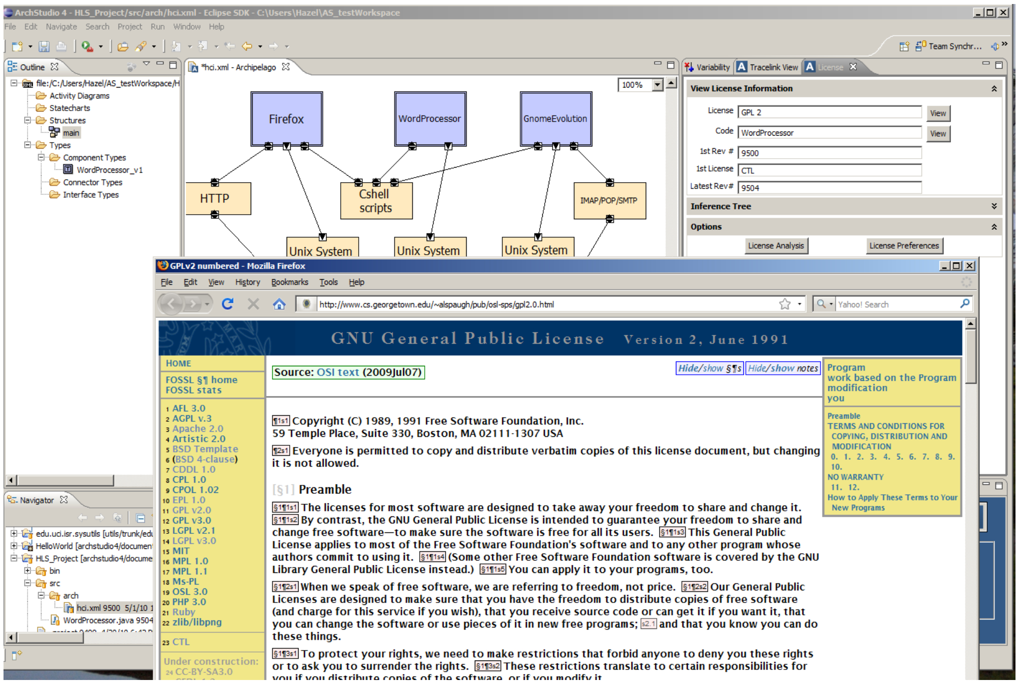
Obligation
PartiallyOrdered
Patient
Right

Classes
Action
Actor
Calculation
Command
Domain
DomainScope
DomainSet
FormulaConjunction
FormulaDisjunction
LicenseAbstract
LicenseConcrete
LicenseConcretePrecursor
LicenseQuantifier
LicenseReference
LicenseSet
LicenseVirtual
Modality
ModalityObligation
ModalityRight
ObligationAbstract
ObligationConcrete

Method Summary

`Action`	**`action`**`()` The right's action.
`Actor`	**`actor`**`()` The right's actor.
`int`	**`compareTo`**`(Named _o)` Compares this object with the specified object in a total ordering by `name()`.
`boolean`	**`conflictsWith`**`(ObligationAbstract _oa)` True if this right conflicts with _obligation.
`ObligationConcrete`	**`correlative`**`()` The correlative obligation.
`static RightConcrete`	**`factory`**`(Actor _actor, ModalityRight _modality, Action _action, PatientConcrete _patient, LicenseConcrete _license)` Returns the concrete right that has the given arguments, constructing it if necessary.
`static RightConcrete`	**`factory`**`(RightAbstract _abstractRight, PatientConcrete _context)` Returns a concrete right subsuming an abstract right.
`LicenseConcrete`	**`license`**`()` The right's license argument.
`Modality`	**`modality`**`()` The right's modality (a `ModalityRight`).
`String`	**`name`**`()` The name (same as `toString()`) used in comparisons.
`protected boolean`	**`overlapsTuple`**`(license3.Tuple _b)` True iff `subsumes(_b)` or `_b.subsumes(this)`.
`PatientConcrete`	**`patient`**`()` The right's patient argument.
`boolean`	**`reserved`**`()` True if this right overlaps with a reserved right.
`RightAbstract`	**`reservedRight`**`()`

License review during license analysis



View License Information

License: GPL 2 [View]
Code: WordProcessor [View]
1st Rev #: 9500
1st License: CTL
Latest Rev#: 9504

Inference Tree

Options

[License Analysis] [License Preferences]

http://www.cs.georgetown.edu/~alspaugh/pub/osl-sps/gpl2.0.html

GNU General Public License Version 2, June 1991

HOME

FOSSL §¶ home
FOSSL stats

1 AFL 3.0
2 AGPL v.3
3 Apache 2.0
4 Artistic 2.0
5 BSD Template
6 (BSD 4-clause)
7 CDDL 1.0
8 CPL 1.0
9 CPOL 1.02
10 EPL 1.0
11 GPL v2.0
12 GPL v3.0
13 LGPL v2.1
14 LGPL v3.0
15 MIT
16 MPL 1.0
17 MPL 1.1
18 Ms-PL
19 OSL 3.0
20 PHP 3.0
21 Ruby
22 zlib/libpng

23 CTL

Under construction:
24 CC-BY-SA3.0

Source: OSI text (2009Jul07)

¶1s1 Copyright (C) 1989, 1991 Free Software Foundation, Inc.
59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

¶2s1 Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

[§1] Preamble

§1¶1s1 The licenses for most software are designed to take away your freedom to share and change it. §1¶1s2 By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software—to make sure the software is free for all its users. §1¶1s3 This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. §1¶1s4 (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) §1¶1s5 You can apply it to your programs, too.

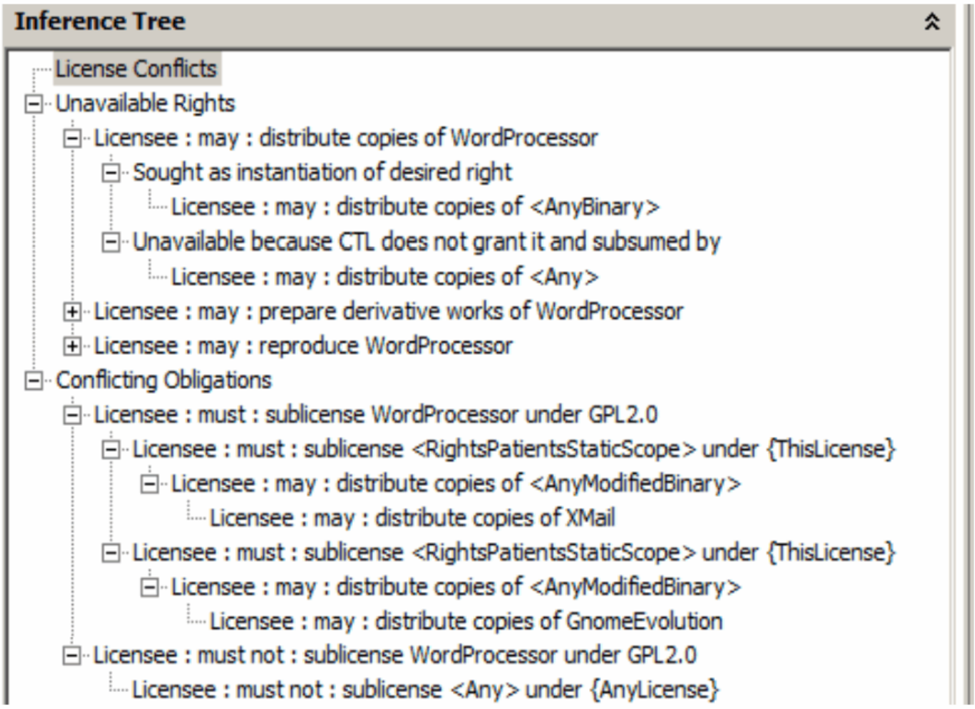
§1¶2s1 When we speak of free software, we are referring to freedom, not price. §1¶2s2 Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; s2.1 and that you know you can do these things.

§1¶3s1 To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. §1¶3s2 These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

Program
work based on the Program
modification
you

Preamble
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION
0. 1. 2. 3. 4. 5. 6. 7. 8. 9. 10.
NO WARRANTY
11. 12.
How to Apply These Terms to Your New Programs

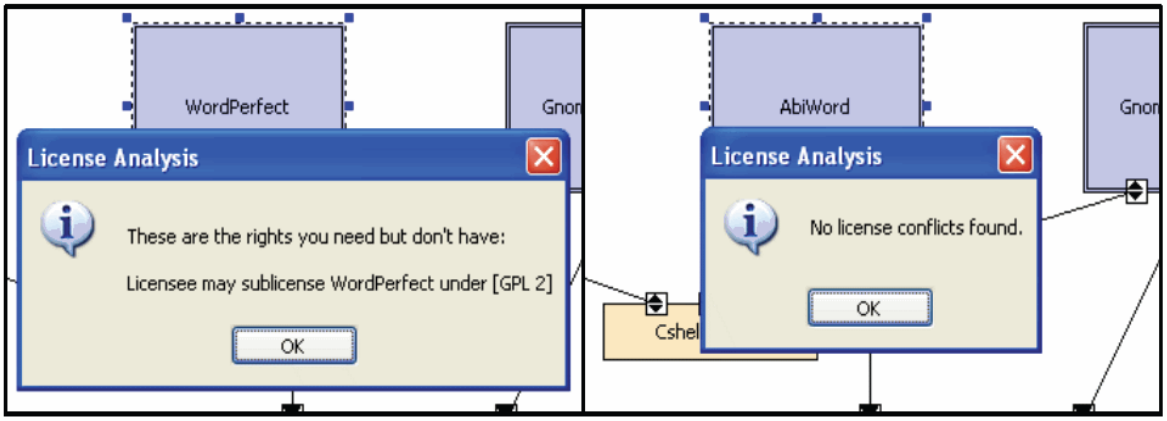
Reasoning structure during analysis



Inference Tree

- License Conflicts
- Unavailable Rights
  - Licensee : may : distribute copies of WordProcessor
    - Sought as instantiation of desired right
      - Licensee : may : distribute copies of <AnyBinary>
    - Unavailable because CTL does not grant it and subsumed by
      - Licensee : may : distribute copies of <Any>
  - Licensee : may : prepare derivative works of WordProcessor
  - Licensee : may : reproduce WordProcessor
- Conflicting Obligations
  - Licensee : must : sublicense WordProcessor under GPL2.0
    - Licensee : must : sublicense <RightsPatientsStaticScope> under {ThisLicense}
      - Licensee : may : distribute copies of <AnyModifiedBinary>
        - Licensee : may : distribute copies of XMail
    - Licensee : must : sublicense <RightsPatientsStaticScope> under {ThisLicense}
      - Licensee : may : distribute copies of <AnyModifiedBinary>
        - Licensee : may : distribute copies of GnomeEvolution
  - Licensee : must not : sublicense WordProcessor under GPL2.0
    - Licensee : must not : sublicense <Any> under {AnyLicense}

Results from license analyses with system component replacement



These are the rights you need but don't have:

Licensee may sublicense WordPerfect under [GPL 2]

No license conflicts found.

Acknowledgements

This material is based upon work supported by the Naval Postgraduate School Acquisition Research Program under Grant No. N00244-14-1-0030. The views expressed in materials or publications, and/or made by the presenters, do not necessarily reflect the official policies of the Naval Postgraduate School, nor does mentions of trade names, commercial practices, or organizations imply endorsement by the U.S. Government.

INSTITUTE *for* SOFTWARE RESEARCH
UNIVERSITY *of* CALIFORNIA · IRVINE

ACQUISITION RESEARCH PROGRAM
GRADUATE SCHOOL OF BUSINESS & PUBLIC POLICY
NAVAL POSTGRADUATE SCHOOL
555 DYER ROAD, INGERSOLL HALL
MONTEREY, CA 93943

www.acquisitionresearch.net